CLUBE

Z80

Maio/85

N.º 32

Coordenadores: ANA PAULA • ALEXANDRE SOUSA • J. MAGALHAES • AV. BOAVISTA, 832-2.º T. • TELEF. 65127 • 4100 PORTO

NESTE NÚMERO

No interior:

Folheto Mercado Z80

Edição: Clube Z80

Fotocomposição: Fotomecânica Mabreu/Porto

Impressão: Ramos dos Santos & C.ª, Lda./Porto

Tiragem: 500 exemplares, Maio 1985

INTRODUÇÃO AO CÓDIGO MÁQUINA

Autor: FERNANDO PRECES
SACAVÉM

(Cont. dos números anteriores)

Ensaio 4 — **(Spectrum) Rotação de 90°**

Esta rotina efectua uma rotação de 90° à posição inicial de qualquer caractere UGD, (caractere gráfico definido pelo utilizador).

Exemplo:

Fig. 1

Fig. 2

Fig. 3

Fig. 4

```
            ORG 23300          ; para RAMTOP um 65300
            LD HL (U.G.D.)     ; variável 23675
            LD E 128
NEWBIT:     PUSH HL            ; byt seguinte (parâmetros
            LD C 0               iniciais
            LD B 1
NEWBYTE:    LD A E             ; byte seguinte
            AND (HL)
            CP 0
            JR Z NOSET
            LD A C
            ADD A B
            LD C A
NOSET:      SLA B
            INC HL
            JR NC NEWBYTE
            POP HL
            PUSH BC
            SRL E
            JR NC NEWBIT
            LD DE 7
            ADD HL DE
            LD B 8             ; contador
REPLACE:    POP DE
            LD (HL) E
            DEC HL
            DJNZ REPLACE
            RET
```

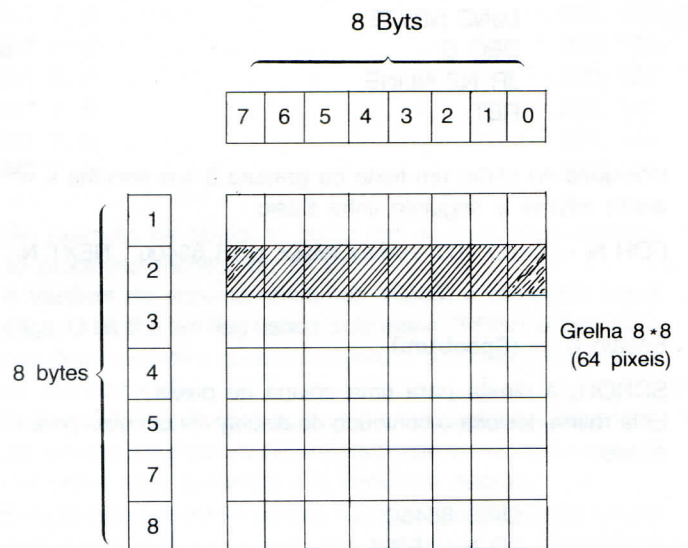
O Basic para a experiência:

FOR N = 1 TO 4: PRINT PRINT " A (Gráfico) " ; : RANDO MIZE USR 23300 : NEXT N

Neste ensaio obtem a imagem das 3 rotações indicadas pelas figs. 2 a 4.

Como funciona a rotina:

Já sabemos que um caractere é formado por um grupo de 64 pixeis (8 * 8) e que cada linha horizontal de 8 pixeis forma um byte.

Exemplo:

Para simplificar a explicação, em vez da nave (figs. 1 a 4), vamos supor que o caractere original é. constituído por uma única barra escura do nível do segundo byte. A sua composição numerica será então a seguinte:

O segundo byte do caractere igual ao binário 11111111 e os restantes iguais ao binário 0.

Após a 1.ª rotação (90° no sentido dos ponteiros do relógio), teremos:
Todos os bytes iguais ao binário 00000010.

Após a 2.ª, teremos:

O byte 7 igual ao binário 11111111 e os restantes iguais ao binário 0.

Depois da 3.ª, teremos:

Todos os bytes iguais ao binário 01000000.

Analizando estas formações, podemos verificar que numa rotação de 90°, os bits (7) de cada um dos 8 bytes que compõem o caractere, formarão o futuro byte (1), e que os bits (6), o byte (2) e assim sucessivamente.
É este de facto o trabalho real da rotina apresentada acima. Os novos bytes são depois armazenados no Stack e recolhidos a seguir pela acção dum contador (REPLACE) que os coloca nos respectivos endereços, neste caso o caractere A (U.G.D.).

Ensaio 5 — **(Spectrum)**

SCROLL à esquerda para uma coluna de pixeis.
Esta rotina desloca o conteúdo do display file um pixel para a esquerda.

```
        ORG 65400
        LD HL 22527
        LD C 192
NLINE   LD B 32
        OR A
NBYTE   RL (HL)
        DEC HL
        DJNZ NBYTE
        DEC C
        JR NZ NLINE
        RET
```

Introduza no ecran um texto ou gravura à sua escolha e no ecran inferior a seguinte linha Basic:

FOR N = 0 TO 100: RANDOMIZE USR 65400 : NEXT N

Ensaio 6 — **(Spectrum)**

SCROLL à direita para uma coluna de pixeis.
Esta rotina desloca o conteúdo do display file um pixel para a direita.

```
        ORG 65450
        LD HL 16384
        LD C 192
NLINE   LD B 32
        OR A
NBYTE   RR (HL)
        INC HL
        DJNZ NBYTE
        DEC C
        JR NZ NLINE
        RET
```

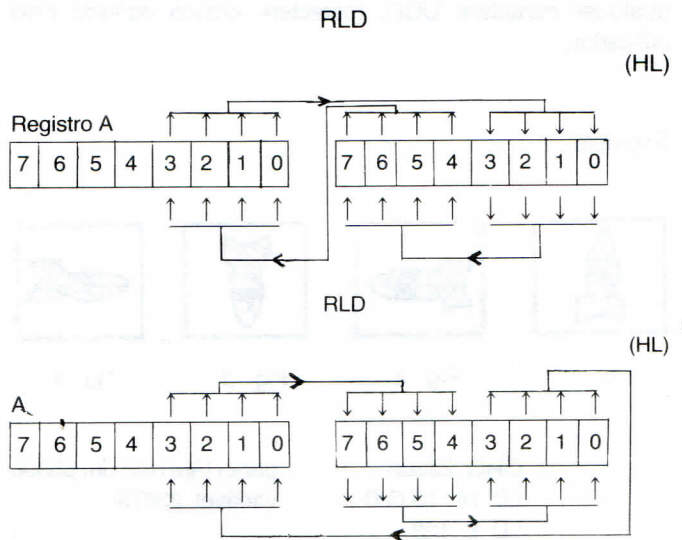
Como no ensaio anterior, escreva algo no ecran superior e no inferior introduza esta linha Basic:

FOR N = 0 TO 100: RANDOMIZE USR 65450 : NEXT N

Nota: Repare que estas duas rotinas não deslocam os atributos. No entanto, são importantes para estudo e como treino das instruções de rotação.

Subgrupo C — **As instruções RLD e RRD**

Dar um formato decimal à codificação binária foi a seu tempo um passo, que os fabricantes de microprocessadores decidiram introduzir nos respectivos Assembleres, dado um certo entusiasmo da parte de alguns **génios** da programação.
Contudo a ideia fracassou pois o sistema em si, é uma constante fonte de erros.
Estas duas instruções foram concebidas para permutar entre o acumulador e o registo HL, 4 bits em formato decimal.

A notação decimal em codificação binária (BCD) emprega apenas 4 bites dos 8 existentes em cada celula. E dessa 1/2 celula, nem todas as permutações possíveis são aproveitadas. Pode por aqui o leitor imaginar a extensão de memória necessária para trabalhar em tal sistema.
A título de informação apresento nas duas figuras em baixo, a sequência de trocas de bits entre os 2 mencionados registros, efectuadas por estas instruções.

Grupo 14 — **As instruções de manejo do bit**

Por vezes não tratamos um byte no seu conjunto, tal como uma palavra de **8 bits,** mas como um agrupamento de bits independentes, usados como flags. Vimos no registro F os seus bits serem utilizados separadamente uns dos outros e a forma como cada um nos transmite algo, sobre uma operação, uma comparação, etc.
Sabemos também que no Ficheiro de Atributos, um byte é utilizado duma forma semelhante pois cada um dos seus bits indicam características em termos de cor, de brilho, etc.
Este grupo de instruções permite testar o **estado** dum específico bit, no interior de qualquer célula de registro (excepto do F), e ainda a possibilidade de o modificar sem **alterar o valor dos restantes.** Posso afirmar que sabendo aplica-las, um fantástico campo de manobra é colocado à disposicão do programador.

Subgrupo A — **As instruções BIT**

Estas instruções testam o bit especificado dentro dum registro também indicado. O resultado do teste é colocado no **zero flag.**

Se o valor testado for **0**, o flag é passado a **1**
« « « « **1**, « « **1**

Mnemónicas	**Códigos**
BIT 0, (HL)	203, 70
BIT 0, (IX + d)	221, 203, + d, 70
BIT 0, (IY + d)	253, 203, + d, 70
BIT 0, A	203, 71
BIT 0, B	203, 64
BIT 0, C	203, 65
BIT 0, D	203, 66

BIT 0, E	203, 67
BIT 0, H	203, 68
BIT 0, L	203, 69
BIT 1, (HL)	203, 78
BIT 1, (IX + d)	221, 203, + d, 78
BIT 1, (IY + d)	253, 203, + d, 78
BIT 1, A	203, 79
BIT 1, B	203, 72
BIT 1, C	203, 73
BIT 1, D	203, 74
BIT 1, E	203, 75
BIT 1, H	203, 76
BIT 1, L	203, 77
BIT 2, (HL)	203, 86
BIT 2, (IX + d)	221, 203, + d, 86
BIT 2, (IY + d)	253, 203, + d, 86
BIT 2, A	203, 87
BIT 2, B	203, 80
BIT 2, C	203, 81
BIT 2, D	203, 82
BIT 2, E	203, 83
BIT 2, H	203, 84
BIT 2, L	203, 85
BIT 3, (HL)	203, 94
BIT 3, (IX + d)	221, 203, + d, 94
BIT 3, (IY + d)	253, 203, + d, 94
BIT 3, A	203, 95
BIT 3, B	203, 88
BIT 3, C	203, 89
BIT 3, D	203, 90
BIT 3, E	203, 91
BIT 3, H	203, 92
BIT 3, L	203, 93
BIT 4, (HL)	203, 102
BIT 4, (IX + d)	221, 203, + d, 102
BIT 4, (IY + d)	253, 203, + d, 102
BIT 4, A	203, 103
BIT 4, B	203, 96
BIT 4, C	203, 97
BIT 4, D	203, 98
BIT 4, E	203, 99
BIT 4, H	203, 100
BIT 4, L	203, 101
BIT 5, (HL)	203, 110
BIT 5, (IX + d)	221, 203, + d, 110
BIT 5, (IY + d)	253, 203, + d, 110
BIT 5, A	203, 111
BIT 5, B	203, 104
BIT 5, C	203, 105
BIT 5, D	203, 106
BIT 5, E	203, 107
BIT 5, H	203, 108
BIT 5, L	203, 109
BIT 6, (HL)	203, 118
BIT 6, (IX + d)	221, 203, + d, 118
BIT 6, (IY + d)	253, 203, + d, 118
BIT 6, A	203, 119
BIT 6, B	203, 112
BIT 6, C	203, 113
BIT 6, D	203, 114
BIT 6, E	203, 115
BIT 6, H	203, 116
BIT 6, L	203, 117
BIT 7, (HL)	203, 126
BIT 7, (IX + d)	221, 203, + d, 126
BIT 7, (IY + d)	253, 203, + d, 126
BIT 7, A	203, 127
BIT 7, B	203, 120
BIT 7, C	203, 121
BIT 7, D	203, 122
BIT 7, E	203, 123
BIT 7, H	203, 124
BIT 7, L	203, 125

Um exemplo da aplicação deste tipo de instruções, extraído do programa monitor do ZX 81.

A variável do sistema, endereço 16385, é composta por 8 flags. O bit 0 é um flag usado pela rotina "PRINT o Teclado", que determina se é ou não necessário um espaço extra antes de um comando.

Todos nós já reparamos que após a projecção de um comando, um espaço é automaticamente criado, servindo de separador entre esse comando e o caractere seguinte.

Este tipo de separador porem, não é criado pela rotina citada, pois neste caso (a projecção das palavras de comando) o ZX sabe que é necessário um espaço. Mas vejamos este exemplo:

10 IF **A** THEN GOTO B

Nesta linha encontram-se 3 comandos. IF, THEN e GOTO. entre A, uma variável, e o comando THEN, o ZX só reconhece que é necessário um espaço quando o comando é primido, alterando o flag de imediato.

As linhas a seguir mostram-nos o teste do flag.

```
2385    BITO, (IY + 1)     ; testa o flag
2389    JR NZ, 2393        ; se a tecla primida for
                             comando não salta.
2391    XOR A              ; limpa A
2392    RST 16             ; projecta um espaço

2393    LD A, (BC)         ; tecla seguinte.
...
```

Nota 1: O registro IY é apontado pela rotina de iniciação para a 1.ª variável do sistema (16384).

Nota 2: É natural que o leitor se interrogue da razão porque até agora, como exemplos de aplicação sobre a ROM, eu apenas tenha citado o ZX 81. Como resposta direi que a primeira parte do 3.º capítulo é dedicado à ROM do Spectrum.

Subgrupo B — **As instruções RES**

Estas instruções permitem colocar um bit de um registro, no valor 0 (RESET). Se o respectivo bit já se encontra a 0, tudo se mantem.

Mnemónicas	Códigos
RES 0, (HL)	203, 134
RES 0, (IX + d)	221, 203, + d, 134
RES 0, (IX + d)	253, 203, + d, 134
RES 0, A	203, 135
RES 0, B	203, 128
RES 0, C	203, 129
RES 0, D	203, 130
RES 0, E	203, 131
RES 0, H	203, 132
RES 0, L	203, 133
RES 1, (HL)	203, 142
RES 1, (IX + d)	221, 203, + d, 142
RES 1, (IY + d)	253, 203, + d, 142
RES 1, A	203, 143
RES 1, B	203, 136
RES 1, C	203, 137
RES 1, D	203, 138
RES 1, E	203, 139
RES 1, H	203, 140
RES 1, L	203, 141
RES 2, (HL)	203, 150
RES 2, (IX + d)	221, 203, + d, 150
RES 2, (IY + d)	253, 203, + d, 150
RES 2, A	203, 151
RES 2, B	203, 144
RES 2, C	203, 145
RES 2, D	203, 146
RES 2, E	203, 147
RES 2, H	203, 148
RES 2, L	203, 149
RES 3, (HL)	203, 158
RES 3, (IX + d)	221, 203, + d, 158
RES 3, (IY + d)	253, 203, + d, 158
RES 3, A	203, 159
RES 3, B	203, 152
RES 3, C	203, 153
RES 3, D	203, 154
RES 3, E	203, 155
RES 3, H	203, 156
RES 3, L	203, 157
RES 4, (HL)	203, 166
RES 4, (IX + d)	221, 203 + d, 166
RES 4, (IY + d)	253, 203, + d, 166
RES 4, A	203, 167
RES 4, B	203, 160
RES 4, C	203, 161
RES 4, D	203, 162
RES 4, E	203, 163
RES 4, H	203, 164
RES 4, L	203, 165
RES 5, (HL)	203, 174
RES 5, (IX + d)	221, 203, + d, 174
RES 5, (IY + d)	253, 203, + d, 174
RES 5, A	203, 175
RES 5, B	203, 168
RES 5, C	203, 169
RES 5, D	203, 170
RES 5, E	203, 171
RES 5, H	203, 172
RES 5, L	203, 173
RES 6, (HL)	203, 182
RES 6, (IX + d)	221, 203, + d, 182
RES 6, (IY + d)	253, 203, + d, 182
RES 6, A	203, 183
RES 6, B	203, 176
RES 6, C	203, 177
RES 6, D	203, 178
RES 6, E	203, 179
RES 6, H	203, 180
RES 6, L	203, 181
RES 7, (HL)	203, 190
RES 7, (IX + d)	221, 203, + d, 190
RES 7, (IY + d)	253, 203, + d, 190
RES 7, A	203, 191
RES 7, B	203, 184
RES 7, C	203, 185
RES 7, D	203, 186
RES 7, E	203, 187
RES 7, H	203, 188
RES 7, L	203, 189

Este tipo de instruções são muito usadas para alterar o valor dum flag, ou as cores de um écran.
Um exemplo extraído do programa monitor do ZX81, mostra-nos como é usada uma instrução RES na rotina de comando FAST.
Os bits 6 e 7 da variável do sistema 16443 (CD FLAG), controlam a operação dos modos 'SLOW' e 'FAST'.

Endereços	Mnemónicas	Comentários
3875	CALL 743	Rotina «SET FAST MODE»
3878	RES 6, (IX + 59)	bit 6 a zero, na variável 16443
3882	RET	Retorno em «FAST MODE»
		Subrotina «SET FAST MODE»
743	BIT 7, (IX + 59)	teste ao bit 7 da variável 16443
747	RET Z	Salto p/ 3878 se o bit 7 for zero
748	HALT	faz uma Pausa. Instruções
749	OUT (254), A	ainda a estudar.
751	RES 7, (IX + 59)	bit 7 a zero, na variável 16443
755	RET	retorno para o endereço 3878

SUBGRUPO C — As instruções SET

Este tipo de instruções permitem colocar o bit referenciado, no valor 1. Se esse bit já for 1 a instrução confirmará apenas a situação.

Mnemónicas	Códigos
SET 0, (HL)	203, 198
SET 0, (IX + d)	221, 203, + d, 198
SET 0, (IY + d)	253, 203, + d, 198
SET 0, A	203, 199
SET 0, B	203, 192
SET 0, C	203, 193

SET 0, D	203, 194
SET 0, E	203, 195
SET 0, H	203, 196
SET 0, L	203, 197
SET 1, (HL)	203, 206
SET 1, (IX + d)	221, 203, + d, 206
SET 1, (IY + d)	253, 203, + d, 206
SET 1, A	203, 207
SET 1, B	203, 200
SET 1, C	203, 201
SET 1, D	203, 202
SET 1, E	203, 203
SET 1, H	203, 204
SET 1, L	203, 205
SET 2, (HL)	203, 214
SET 2, (IX + d)	221, 203, + d, 214
SET 2, (IY + d)	253, 203, + d, 214
SET 2, A	203, 215
SET 2, B	203, 208
SET 2, C	203, 209
SET 2, D	203, 210
SET 2, E	203, 211
SET 2, H	203, 212
SET 2, L	203, 213
SET 3, (HL)	203, 222
SET 3, (IX + d)	221, 203, + d, 222
SET 3, (IY + d)	253, 203, + d, 222
SET 3, A	203, 223
SET 3, B	203, 216
SET 3, C	203, 217
SET 3, D	203, 218
SET 3, E	203, 219
SET 3, H	203, 220
SET 3, L	203, 221
SET 4, (HL)	203, 230
SET 4, (IX + d)	221, 203, + d, 230
SET 4, (IY + d)	253, 203, + d, 230
SET 4, A	203, 231
SET 4, B	203, 224
SET 4, C	203, 225
SET 4, D	203, 226
SET 4, E	203, 227
SET 4, H	203, 228
SET 4, L	203, 229
SET 5, (HL)	203, 238
SET 5, (IX + d)	221, 203, + d, 238
SET 5, (IY + d)	253, 203, + d, 238
SET 5, A	203, 239
SET 5, B	203, 232
SET 5, C	203, 233
SET 5, D	203, 234
SET 5, E	203, 235
SET 5, H	203, 236
SET 5, L	203, 237
SET 6, (HL)	203, 246
SET 6, (IX + d)	221, 203, + d, 246
SET 6, (IY + d)	253, 203, + d, 246
SET 6, A	203, 247
SET 6, B	203, 240
SET 6, C	203, 241
SET 6, D	203, 242
SET 6, E	203, 243
SET 6, H	203, 244
SET 6, L	203, 245
SET 7, (HL)	203, 254
SET 7, (IX + d)	221, 203, + d, 254
SET 7, (IY + d)	253, 203, + d, 254
SET 7, A	203, 255
SET 7, B	203, 248
SET 7, C	203, 249
SET 7, D	203, 250
SET 7, E	203, 251
SET 7, H	203, 252
SET 7, L	203, 253

Também estas instruções são muito utilizadas para alterar bits de flags, mudar atributos duma imagem, etc.
Um exemplo extraído no monitor do ZX81, mostra-nos como uma instrução SET vai alterar o bit 6 do flag CD, variável 16443, de modo a comutar a projecção da imagem do Modo FAST para o Modo SLAW.
Na rotina do comando SLAW, o bit 6 do flag é colocado no valor 1, accionando um salto para a rotina de Projecção e comutando o sistema para o Modo SLAW. Nada acontece se o sistema já estiver no Modo SLAW.

Endereços	Mnemónicas	Comentários
3883	SET, 6 (IX + 59)	Coloca em 1 bit 6 de 16443
3887	JP 519	Salto para o Display routine

Ensaio 1 — SPECTRUM (Mudança instantânea da cor da tinta num écran).

```
        ORG 65500
        LD HL 22528     ; início dos Atributos
        LD BC 704       ; extensão
        LD E 6          ; cor pretendida
LOOP;   LD A (HL)
        RES 0, A
        RES 1, A
        RES 2, A        ; RES aos 3 bits de tinta
        ADD A E         ; soma a nova cor
        LD (HL) A       ; devolve o atributo
        INC HL          ; novo atributo
        DEC BC          ; – 1 no contador
        LD A C
        CP 0
        JR NZ LOOP      ; salto se C < > 0
        LD A B
        CP 0
        JR NZ LOOP      ; idem
        RET
RANDOMIZE USR 65500
```

Nota: Pode o leitor, para treino, introduzir pequenas modificações ao programa deste ensaio, de modo a alterar cores do Papel ou a tonalidade da cor através do brilho, sobre uma imagem guardada em memória ou projectada no écran. Experimente.

Continua no próximo número

NEW BRAIN

Traduzido e adaptado do PCN (vol. 1 n.º 37)
por Isabel Cristina — LOG

Esta rotina permite obter uma string de qualquer comprimento sem imprimir o sinal da continuação. Também apagará qualquer palavra que não esteja completa numa linha e coloca-la-á na linha seguinte.

Note que a tecla de espaço é carregada uma vez entre aspas na linha 1020 e também uma vez no fim da string antes de accionar NL. Para usar a rotina num display de 80 colunas, o 40 da linha 390 deverá ser substituído por 80.

```
390 C = 40:PRINT"INPUT STRING BELOW."
400 PUT 10:LINPUT (" ") A$:PUT 31
420 L = LEN (A$)
440 FOR I = 1 TO L
460 X = C + J
480 D = J + 1
500 GOSUB 1000
520 ON ERROR GOTO 2000
560 PRINT MID$(A$,D,J-D)
580 NEXT I
1000 FOR J = X TO 1 STEP-1
1020 IF MID$(A$,J,1) = " " THERN RETURN
1040 NEXT J
1060 RETURN
2000 END
```

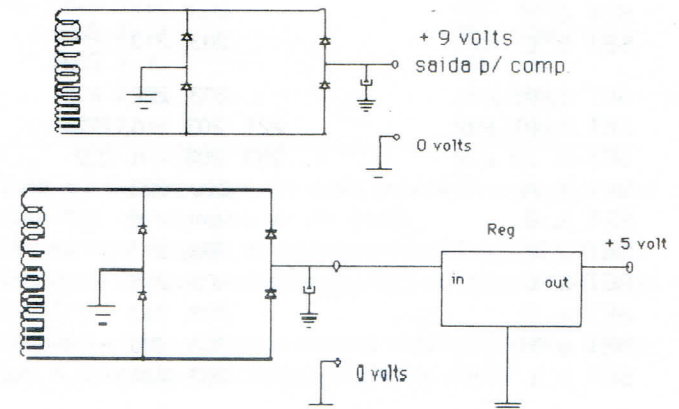
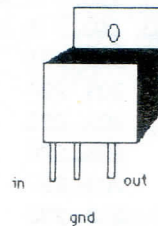
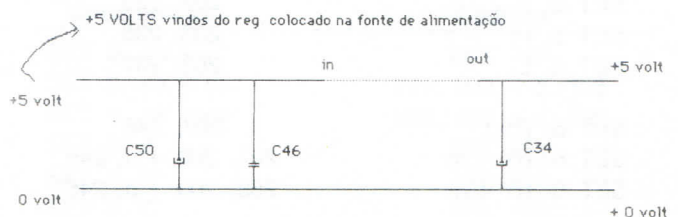
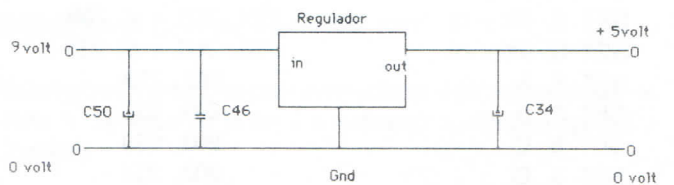
HARDWARE

Autor: PAULO METELO

O problema do aquecimento no Spectrum, que em muitos casos é "excessivo" e digamos que não será um calor que os integrados se dêem muito bem e este calor que se sente bastando para tal pôr a mão em cima do computador, provém somente de um único sítio que no caso, é o regulador de 5 volts o qual tem um dissipador que depende de versão para versão, pois nalgumas versões o dissipador fica situado na parte superior e em outras versões o mesmo fica situado na parte lateral direita, e isto tudo por cima do circuito impresso e dos seus respectivos componentes.

Ora para evitar que este calor circule dentro do computador poder-se-ia pôr um pequeno ventilador na parte traseira do computador mas tal iria tornar-se um pouco caro e bastante trabalhoso, por isso aqui vai a maneira que sai bastante económico e não provocará nenhum calor dentro do computador. Para tal basta instalar dentro da própria fonte de alimentação o regulador que se encontra dentro do computador.

Para tal basta retirar o regulador e o respectivo dissipador dentro do computador e em seguida curtucircuitar os sítios onde estavam os pinos (in) e (out) do regulador, e na fonte de alimentação desliga-se o cabo de alimentação do positivo e intercala-se o regulador entre o circuito e o cabo, ligando ao cabo o pino (out) e o pino (in) ao circuito não esquecendo de ligar o pino (GND) a massa da fonte de alimentação.

Com um pouco de habilidade deverá por um dissipador no regulador para que este dissipe um pouco o calor, isto tudo terá de ser instalado dentro da fonte de alimentação por isso deverá estudar o espaço desta para depois poder trabalhar mais á vontade, porque o espaço existente e como depois poderá ver é bastante pequeno, por isso tenha paciência e trabalhe com cuidado.

Circuito de alimentação com o regulador incluido

CIRCUITO ORIGINAL

ATENUADORES

Um dos circuitos regularmente usados em electrónica é o Atenuador Resistivo. Na sua forma mais simples, usa três, quatro ou cinco resistências e a sua principal aplicação tem a ver com a tentativa de reduzir um sinal ou equilibrar dois círculos de impedâncias distintas.

Existem dois tipos de atenuadores mais comuns, os tipos T e PI, cada um dos quais pode ser construído com configuração equilibrada ou desequilibrada.

Em circuito de baixa potência, não se colocam grandes problemas em termos de escolha a fazer, no entanto, se tratamos de níveis elevados de potência, a dissipação das resistências deve ser tomada em consideração. Nessas circunstâncias, devemos escolher a configuração «T» dado que no esquema em «PI» a maioria da potência de entrada é absorvida por uma só resistência, enquanto na forma «T» será repartida pela resistência de entrada e pela resistência central em paralelo.

Uso do Programa

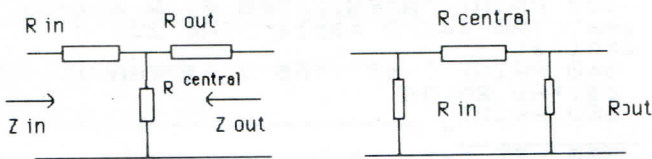
O programa calcula as resistências necessárias para construir as impedâncias que equilibram as redes T ou PI, partindo de valores de atenuação pré-determinados e imprime os resultados apresentando-os em forma de tabela.
A primeira acção do programa após o RUN, será a de apresentar o MENU que permite seleccionar a rede desejada.

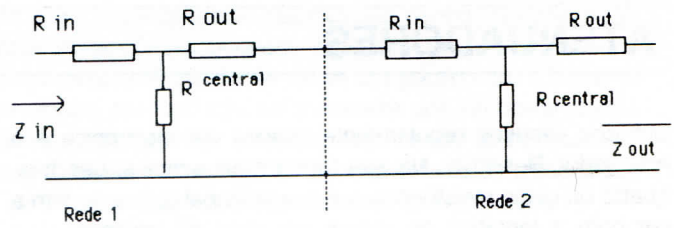
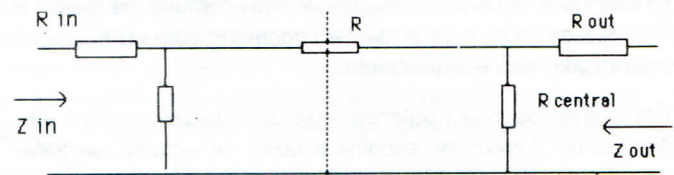
1) Desequilibrada em T
2) Desequilibrada em PI
3) Equilibrada em T
4) Equilibrada em PI

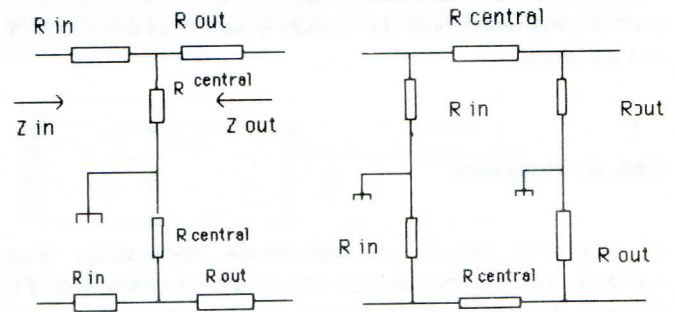
Seleccionada a opção da rede, terá de fornecer ao programa o valor mais baixo em dB requerido para a atenuação, o passo de construção da tabela e o valor mais alto (atenuação).

Quando não for possível providenciar uma impedância definida com o valor solicitado, aparecerá uma mensagem «Configuração impossível».

```
   1 BORDER 0: PAPER 0: INK 7
  10 REM Programa p/calculo de A
tenuadores
  30 LET B$="EQUILIBRADA"
  40 LET N$=" REDE "
  50 LET O$=" OHMS "
  60 DEF FN A(R)=INT (10*R+.5)/1
0
  70 CLS
  80 PRINT
  90 PRINT TAB 3;"PROJECTO DE RE
DE 'ATENUADORES'"
 100 PRINT
 110 FOR I=1 TO 4
 120 GO SUB (560+(I-1)*20)
 130 NEXT I
 140 PRINT TAB 3;"seleccionar ti
po de REDE (1...4"
 150 INPUT d
 160 IF INT (D)<>D OR D<1 OR D>4
 THEN GO TO 70
 170 CLS
 180 PRINT
 190 PRINT "valores Impedancia I
N,OUT";o$
 200 INPUT zi,zo
 210 IF zo>0 AND zi>0 THEN GO TO
 240
 220 PRINT "impedancia deve ser
positiva"
 230 GO TO 180
 240 PRINT
 250 PRINT "valor dB (gama) Low,
Step,High"
 260 INPUT a1,a2,a3
 270 CLS
 280 PRINT
 290 GO SUB (560+(I-1)*20)
 300 PRINT
 310 PRINT "Zin= ";zi;o$;" Zout=
";zo;o$
 320 PRINT
 330 PRINT "ATEN.";TAB 6;"R entr
ada";TAB 14;"R saida";TAB 23;"R
central"
 340 PRINT " dB";TAB 7;o$;TAB 15
;o$;TAB 25;o$
 350 PRINT "______________________
__________"
 360 LET r=zi/zo
 370 IF d<3 THEN GO TO 400
 380 LET zi=zi/2
 390 LET zo=zo/2
 400 FOR a=a1 TO a3 STEP a2
 410 IF a=0 THEN GO TO 490
 420 LET n=SQR (10↑(a/10)*r)
 430 LET h=n↑2+r
 440 LET i=n↑2-r
 450 IF d=1 OR d=3 THEN GO SUB 6
60: GO TO 465
 460 IF d=2 OR d=4 THEN GO SUB 7
00
 465 IF r1<0 OR r2<0 OR r3<0 THE
N GO TO 490
 470 PRINT ;a;TAB 8;FN a(r1);TAB
 16;FN a(r2);TAB 25;FN a(r3)
 480 GO TO 500
 490 PRINT ;a;TAB 6;"Configuraca
o Impossivel"
 500 NEXT a
 510 PRINT
 520 PRINT "correr o Programa ?
(s/n)"
 530 INPUT a$: IF a$="s" OR a$="
S" THEN GO TO 70
 550 STOP
 560 PRINT "1. ";N$;"DES";B$;" -
T -"
 570 GO TO 640
 580 PRINT "2. ";N$;"DES";B$;" -
PI -"
 590 GO TO 640
 600 PRINT "3. ";N$;B$;" - T -"
 610 GO TO 640
 615 RETURN
 620 PRINT "4. ";N$;B$;" - PI -"
 630 PRINT
 640 RETURN
 660 LET r3=zi*n*2/i
 670 LET r2=zo*h/i-r3
 680 LET r1=zi*h/i-r3
 690 RETURN
 700 LET r3=zo*i/2/n
 710 LET r2=1/(h/i/zo-1/r3)
 720 LET r1=1/(h/i/zi-1/r3)
 730 RETURN
```

```
     PROJECTO DE REDE 'ATENUADORES

   1.   REDE DESEQUILIBRADA - T -
   2.   REDE DESEQUILIBRADA - PI -
   3.   REDE EQUILIBRADA - T -
   4.   REDE EQUILIBRADA - PI -

     seleccionar tipo de REDE (1..
  .4

  Zin= 300 OHMS   Zout= 75 OHMS

  ATEN.  R entrada
                R saida    R central
   dB      OHMS     OHMS       OHMS

  5        Configuracao Impossivel
  10     . Configuracao Impossivel
  15          907      85.1     408.4
  20          486.9    81.6     742.5
  25          384.3    79       1329.5
  30          342.7    77.3     2369.3
  35          322.8    76.3     4216.2
  40          312.4    75.7     7499.3

  correr o Programa ? (s/n)

  os valores do teste sao: dB rang
  e Low=5, Step=5,High=40
```

Rede desiquilibrada tipo T (esquerda)

" " " PI (direita)

Combinação de duas REDES tipo T

R = R out (rede 1) + R in (rede 2)

Combinação de duas REDES tipo T COMPLETA

Rede equilibrada tipo T

Rede equilibrada tipo PI

A ligação a terra (massa) pode ser omitida se o quiser

AUMENTAR A VELOCIDADE NO SPECTRUM

As rotinas que passamos a expor permitem poupar tempo, em relação ao processamento dos programas que usa no seu Spectrum.

Pode enriquecer os seus programas, especialmente os de uso profissional, de modo a diminuir a lentidão de tratamento dos GO TO, GOSUB, ciclos FOR NEXT, usar uma matriz em vez de variáveis tipo string e ainda uma rotina para a entrada de dados.

1) GO TO ... estas linhas de programa são sempre lentas e se estiverem incluídas em ciclos FOR/NEXT ainda introduzem maior lentidão. O objectivo de aumentar a rapidez é conseguido, se mudar o valor da variável do sistema denominado PROG.

Usamos duas rotinas em código máquina 'PROGNEXT' e 'PROGBACK'.

ROTINA PROGNEXT/PROGBACK

```
300 LET PROGNEXT = USR "a"
3010 LET PROGBACK = PROGNEXT + 10
3030 RESTORE 3050
3040 FOR I = PROGNEXT TO PROGNEXT + 44 : READ a :
POKE i,a : NEXT I
3050 DATA  42,85,92,34,83,92,1,0,0,201,33,16,92,175,6,19,
     190,48,1,126,35,35
3060 DATA  16,248,198,4,79,42,79,92,9,126,254,128,32,6,
     35,126,254,40,56,217,195,236,27
```

Prognext coloca a variável PROG no endereço apontado pela próxima linha de prgrama.

Progback executa o Reset da mesma variável de forma a assumir o valor normal, o qual e calculado a partir das variáveis STREAMS e CHANS.

Se eventualmente activou um canal, pode ter a informação STATEMENT LOST, que significa código de erro.

Ambas as rotinas são relocatáveis em qualquer ponto acima do RAMTOP.

A próxima listagem mostra um uso típico de activar dados para correr de novo um programa:

```
9900 REM activar os dados existentes a partir da linha 9940
9920 RESTORE USR prognext : REM o computador entende
     que o programa começa na linha 9925
9925 DIM A(100)
9930 FOR I = 1 TO 100 : READ A(I) : NEXT I
9940 DATA ..........
```

Observe agora o uso típico em que é possível acontecer erro enquanto a variável PROG é modificada: existindo um dispositivo de segurança incluído para o caso de algum tentar correr o programa com PROG apontado para a linha 2015:

```
  10 REM declarar as variáveis
  20 DIM A$(31) : LET KEY = 0: LET LEN = 0 : LET CURS = 0
  30 REM ...........
1000 REM
```

```
2000 REM input ate 30 caracteres
2002 REM entrada em 2010 e editar i$
2005 LET i$ = ""
2010 RANDOMISE USR prognext : GO TO 2020 : REM var. PROG aponta para linha
2015
2015 RANDOMISE USR ( USR "a" + 10): RANDOMISE USR 7148 : REM existe um
erro se tentar arrancar daqui. Executa o reset de PROG & relatorio de
STATEMENT LOST
2020 LET a$= i$ : LET len = LEN i$: LET curs= len : PRINT# 0 : REM print#
0 executa o clear do bit 5 das flags
2025 REM
2030 PRINT AT 0,0: ">";a$ (TO curs);"*";a$( curs+1  TO len+1)
2040 IF PEEK 23611 <224 THEN GO TO 2040
2050 POKE 23611, PEEK 23611-32 : LET Key = PEEK 23560 : BEEP .004,40
2056 REM
2060 IF key >= 32 AND Key  <=90 THEN IF len <30 THEN LET a$(curs+1 TO
)=CHR$ key + a$(curs+1 TO) : LET curs = curs +1 M LET len = len +1 :GO TO
2030
2070 IF Key = 12 THEN IF curs THEN LET a$(curs TO) = a$(curs+1 TO) :LET
curs = curs -1 : LET len = len -1 : GOTO 2030
2080 LET curs = curs - (Key=8 AND curs) + (Key =9 AND curs < len)
2095 REM
2100 PRINT AT 0,1: a$(TO len+1)
2110 LET i$=a$ (TO len)
```

2) Quando atribuímos um novo valor a uma variável tipo string, ela e sempre movimentada para o final da lista de variáveis, de modo que a manipulação de strings pode ser lenta. Compara as técnicas que usamos na listagem acima, que torna as entradas de dados muito mais rápidas.

NEW BRAIN — Control de Cassettes

Trad. e Adapt. do PCN (vol 1 n.º 33)
Por: ISABEL CRISTINA — LOG

No NB é possível obter mais de 80 caracteres por linha usando a ficha COMMS e não a ficha PRINTER.
Para isso use // 9 (stream 9) em vez de // 8.
PRINT // 9.... produzirá uma linha impressa ao fim de 80 caracteres.
PUT // 9,... enviará códigos de control para a impressora, conforme as necessidades.
ROTINAS para controlar a cassette:

```
5000 PRINT "CARREGUE NA TECLA REVIEW E DEPOIS
     CARREGUE QUALQUER TECLA PARA CONTI-
     NUAR".
5010 PRINT TAB (10); "CARREGUE NA TECLA * NO FIM
     DA REBOBINAGEM DA CASSETTE PARA CONTI-
     NUAR O PROGRAMA".
5020 GET // 5, CH: ON BREAK GOTO 5030: VERIFY.
5030 ON BREAK GOTO O: RETURN.
```

(Pressupondo que a // 5 (stream 5) foi aberta antes com OPEN // 5,5,.

RESISTÊNCIA EM PARALELO — CÁLCULO

Apresenta-se um pequeno programa que é útil para os entusiastas da electrónica.
Preparado para valores limite entre 10 e 100 ohms (pode ser melhorado pelos leitores) pode calcular o valor de acordo com a tolerância exigida.
A saída obtida na impressora e no écran, corresponde a todas as combinações (dentro de limites práticos) de resistências em paralelo.
Os valores são escolhidos dentro dos valores padrão.

```
   1 BORDER 0: PAPER 0: INK 7
   2 CLS
  20 GO SUB 200
  30 CLS : LET y=0: LET z=0
  40 PRINT "Resistencias Em Para
lelo"
  45 PRINT ''"(valor aprox. de a
cordo c/toler."
  50 INPUT "Valor req.(10...100)
:- ";x,"Tolerancia (%) ? - ";t;"
%"
  60 PRINT AT 2,0;" Valor: ";x,"
Tolerancia:(%) - ";t;"%"
  70 LET t1=1-t/200: LET th=1+t/
200: LET x1=x*t1: LET xh=x*th
  80 FOR n=2 TO 25: IF x>=a(n-1)
AND x<=a(n) THEN LET y=n
  90 LET m=n+24*(x>50): IF x*2>a
(m-1) AND x*2<=a(m) THEN LET z=m
 100 GO TO 110+10*(y=n AND z=m)
 110 NEXT n
 120 PRINT " R(1)    R(2)    (MIN)
SOMA   (MAX)"''
 130 FOR M=Y TO Z: FOR N=Z TO 49
: LET XN=A(N)*A(M)/(A(N)+A(M))
 140 IF XN>XH OR XN<X1 THEN GO T
O 170
 150 PRINT TAB (A(N)<100)+(A(N)<
1E3);A(N);TAB 5+(A(M)<100)+(A(M)
<1E3);A(M);
 160 PRINT TAB 14;INT (XN*T1*100
+.5)/100;TAB 20;INT (XN*100+.5)/
100;TAB 26;INT (XN*TH*100+.5)/10
0
 170 NEXT N: NEXT M
 180 PRINT #0;AT 0,0;"Limites re
q.: ";INT (x*(100-t)+.5)/100;" -
";INT (x*(100+t))/100'"qq tecla
p/obter outro valor"
 190 PAUSE 0: GO TO 30
 200 DIM a(49): FOR n=1 TO 25: R
EAD a(n): LET a(n+24)=a(n)*10: N
EXT n: RETURN
 210 DATA 10,11,12,13,15,16,18,2
0,22,24,27,30,33,36,39,43,47,51,
56,62,68,75,82,91,100
```

```
Resistencias Em Paralelo

 Valor: 55.77    Tolerancia:(%) -
4%lor aprox. de acordo c/toler.
 R(1)    R(2)    (MIN)   SOMA   (MAX)

 470      62     53.68  54.77  55.87
 510      62     54.17  55.28  56.39
 560      62     54.7   55.82  56.94
 620      62     55.24  56.36  57.49
 680      62     55.68  56.82  57.96
 300      68     54.33  55.43  56.54
 330      68     55.25  56.38  57.51
 220      75     54.81  55.93  57.05
 180      82     55.21  56.34  57.46
 150      91     55.51  56.64  57.77
 130     100     55.39  56.52  57.65


 LIMITES REQUERIDOS:           53.54
  -    58
```

CONJUNTO DOS DIVISORES DUM NÚMERO

J. C.

```
#################################
#   CONJUNTO DOS DIVISORES      #
#         DUM NUMERO            #
#################################
          1         123456
          2          61728
          3          41152
          4          30864
          6          20576
          8          15432
         12          10288
         16           7716
         24           5144
         32           3858
         48           2572
         64           1929
         96           1286
        192            643
  5 REM  CONJUNTO DOS DIVISORES
   DUM NUMERO
```

```
  10 CLS
  20 PRINT BRIGHT 1;"###########
######################  CONJUN
TO DOS DIVISORES        ##
DUM NUMERO            ##########
#####################"''
  30 INPUT "NUMERO ? ";N
  40 FOR M=1 TO N
  50 PRINT #0;AT 0,0; BRIGHT 1;M
  60 IF N/M=INT (N/M) THEN BEEP
.2,5: PRINT TAB 11-LEN STR$ M;M;
TAB 26-LEN STR$ (N/M);N/M
  70 IF M>N/M-1 THEN GO TO 90
  80 NEXT M
  90 FOR B=1 TO 10: BEEP .02,5:
NEXT B
 100 PRINT #0; BRIGHT 1;"      PRI
MA<s> PARA TERMINAR"
 110 PAUSE 0
 120 IF INKEY$<>"s" AND INKEY$<>
"S" THEN GO TO 10
 130 STOP
9990 SAVE "DIVISORES" LINE 1
```

NUMERAIS DECIMAIS / Converter Decimal em Extenso

J. M. 85

```
   5 REM    NUMERAIS DECIMAIS
   LINGUAGEM CORRENTE - J.M.85
  10 CLS : DIM z(6)
  20 INPUT "Que numero? ";n
  30 IF n<0 OR n<>INT n THEN PRI
NT "So numeros inteiros,por favo
r!": GO TO 20
  40 IF n>1e10 THEN PRINT ''"nao
 fazes comigo!!!": GO TO 20
  50 GO SUB 9000: PRINT ''n;" e'
";N$: GO TO 20
9000 IF n=0 THEN LET N$="Zero ":
 RETURN
9010 LET N$="": LET milao=INT (n
/1E6): LET milar=n-1E6*milao: IF
 milao=0 THEN GO TO 9040
9020 LET num=milao: GO SUB 9080:
 IF milao=1 THEN LET N$=N$+"milh
ao ": GO TO 9040
9030 LET N$=N$+"milhoes "
9040 IF milar=0 THEN GO TO 9070
9050 LET num=milar: IF milao>0 A
ND milar<1E5 THEN LET N$=N$+"e "
9060 GO SUB 9080
9070 LET N$(1 TO 1)=CHR$ (CODE N
$(1 TO 1)-32): RETURN
9080 LET x5=LEN N$: FOR z=6 TO 1
 STEP -1: LET z(z)=num-INT (num/
10)*10: LET num=INT (num/10): NE
XT z: LET z=1
9090 LET x8=z(z): LET x7=10*z(z+
1)+z(z+2): LET x6=1: IF x8=0 AND
 x7=1 AND z=0 THEN GO TO 9210
9100 LET x6=0: IF x8=0 THEN GO T
O 9140
9110 IF LEN N$<>x5 AND x7=0 THEN
 LET N$=N$+"e "
9120 IF x8=1 AND x7=0 THEN LET N
$=N$+"cem ": GO TO 9210
9130 RESTORE 9327+x8: READ x$: L
ET N$=N$+x$+" "
9140 LET x8=z(z+1): IF x8=0 THEN
 GO TO 9180
9150 IF LEN N$<>x5 THEN LET N$=N
$+"e "
9160 IF x8=1 THEN RESTORE 9300+x
7: READ x$: LET N$=N$+x$+" ": GO
 TO 9210
9170 RESTORE 9318+x8: READ x$: L
ET N$=N$+x$+" "
9180 LET x8=z(z+2): IF x8=0 THEN
 GO TO 9210
9190 IF LEN N$>x5 THEN LET N$=N$
+"e "
9200 RESTORE 9300+x8: READ x$: L
ET N$=N$+x$+" "
9210 IF z>1 THEN RETURN
9220 LET z=z+3: IF LEN N$<=x5 AN
D x6=0 THEN GO TO 9090
9230 IF N$="um " THEN LET N$=""
9240 LET N$=N$+"mil ": GO TO 909
0
9301 DATA "um"
9302 DATA "dois"
9303 DATA "tres"
9304 DATA "quatro"
9305 DATA "cinco"
9306 DATA "seis"
9307 DATA "sete"
9308 DATA "oito"
9309 DATA "nove"
9310 DATA "dez"
9311 DATA "onze"
9312 DATA "doze"
9313 DATA "treze"
9314 DATA "catorze"
9315 DATA "quinze"
9316 DATA "dezasseis"
9317 DATA "dezassete"
9318 DATA "dezoito"
9319 DATA "dezanove"
9320 DATA "vinte"
9321 DATA "trinta"
9322 DATA "quarenta"
9323 DATA "cinquenta"
9324 DATA "sessenta"
9325 DATA "setenta"
9326 DATA "oitenta"
9327 DATA "noventa"
9328 DATA "cento"
9329 DATA "duzentos"
9330 DATA "trezentos"
9331 DATA "quatrocentos"
9332 DATA "quinhentos"
9333 DATA "seiscentos"
9334 DATA "setecentos"
9335 DATA "oitocentos"
9336 DATA "novecentos"
9990 SAVE "EXTENSO" LINE 1
```

DIA DA SEMANA

DIA SEMANA — O programa pede o dia, depois o mês e finalmente o ano com 4 algarismos e informa-nos o dia da semana a que corresponde esse dia.

Só funciona a partir de 1583 pois desde cerca do princípio da nossa era, até esse ano funcionou o calendário Juliano. Em Outubro de 1582, Gregório XIII reformou esse calendário que se encontra em funcionamento ainda nos nossos dias.

```
  10 CLEAR
  20 INK 0
  30 PAPER 6
  40 BORDER 1
  43 PRINT AT 3,2;"PROGRAMA PARA
 DETERMINAR O DIA DA SEMANA DUMA
 DATA (Valido so a partir de 158
3)";AT 15,3;"Para continuar, pre
mir 'C' e depois 'ENTER'": STOP
  46 CLS
  50 INPUT "DIA ? ";A
  60 INPUT "MES ? ";B
  70 INPUT "ANO ? ";C
  72 IF C<1583 THEN PRINT " ESTE
 PROGRAMA SO E VALIDO PARA ANOS
A PARTIR DE 1583": STOP
  80 IF B>2 THEN GO TO 110
  90 LET F=365*C+A+31*(B-1)+INT
 ((C-1/4)-INT .75*INT ((C-1)/100+
 1))
 100 GO TO 120
 110 LET F=365*C+A+31*(B-1)-INT
 (.4*B+2.3)+INT (C/4)-INT (.75*(I
NT (C/100)+1))
 120 LET I=F-(INT (F/7)*7)
 130 PRINT AT 4,2;"O DIA ";A;"/"
;B;"/";C;" CORRESPONDE A"
 140 IF I=1 THEN PRINT " UM
 DOMINGO"
 150 IF I=2 THEN PRINT " UMA
      SEGUNDA FEIRA"
 160 IF I=3 THEN PRINT " UMA
    TERCA FEIRA"
 170 IF I=4 THEN PRINT " UMA
   QUARTA FEIRA"
 180 IF I=5 THEN PRINT " UMA
     QUINTA FEIRA"
 190 IF I=6 THEN PRINT " UMA
     SEXTA FEIRA"
 200 IF I=0 THEN PRINT "UM
      SABADO"
 210 INPUT "PREMIR QUALQUER TECL
A PARA CONTINUAR OU 'N' PARA ACA
BAR E DEPOIS ENTER";K$
 220 IF K$="N" THEN STOP
 230 IF K$<>"N" THEN GO TO 46
```

CONSTRUIR UM PROGRAMA TIPO JOGO DE GUERRA

CONSTRUIR UM PROGRAMA TIPO JOGO DE GUERRA

Acreditamos que conheça as regras do BASIC e a maior parte das suas instruções e comandos, mas será que esta habituado a discernir a forma de as usar?

A prática e o exemplo podem ajudar-vos . . . bastara encontrar uma realização precisa (um jogo por exemplo), definir correctamente as suas regras, e depois resolver os problemas (PACIENTEMENTE) um a um.

Para este exercício nós damos uma série de explicações, se não o satisfizer, então melhore . . . o trabalho e mande-nos a lição em ordem!

O programa proposto, realiza uma espécie de batalha, em termos muito simples, que os leitores podem complicar e melhorar.

Cada jogador dispõe de 9 canhões, alinhados ao alto do campo de combate para o primeiro jogador e em baixo para o segundo. A distância que separa os dois jogadores, é de 7 casas.

Cada jogador deve deslocar o seu canhão de uma só casa: horizontal ou verticalmente. O canhão deslocado, atira automaticamente em linha recta, sobre todas as peças inimigas. As peças atingidas desaparecem do jogo.

Existem casas ditas tipo Montanha que protegem as peças dos tiros inimigos.

Se o jogador não consegue eliminar as peças do inimigo ao final de 40 tiros, podemos avaliar os estragos causados, e aquele que tiver melhor pontuação é considerado vencedor.

ELEMENTOS DO JOGO

Determinemos as diferentes funções do programa necessárias para colocar em acção este jogo de guerra.

É preciso instalar o campo da batalha em memória.
No écran, será necessário gerir o deslocamento dos canhões, obter o tiro e calcular os resultados.

O nosso campo de batalha resume-se a um quadrado de 9 x 9. Dimensionamos portanto uma matriz de 9 por 9, e devemos inicializar os seus valores a zero, com o recurso a dois ciclos, de uma forma clássica . . . . linhas 15 a 19.

O contador de Tiros é inicializado em 1. Após esta primeira etapa, afixamos o terreno de jogo. Vamos usar o modo texto, em que cada casa do écran corresponde a uma coordenada vertical (coluna) que denominamos V, e por uma coordenada horizontal (linha) a que chamamos H.

As linhas de 30 a 32 tratam de situar as casas do jogo, enquanto que as linhas de 50/55 colocam os canhões nas suas posições.

Entretanto não esquecemos que existe uma matriz ainda vazia e que se torna necessário encher de valores. Para cada casa onde se encontra um canhão será necessário guardar a sua presença e a sua potência de tiro.

É isso que nós fazemos nas linhas 60/68, em que as posições da matriz recebem os valores apropriados.

Observemos o que diz respeito a cada canhão:

— presença .... (casa diferente de zero).
— potência de tiro .... (parte inteira do seu valor).
— o seu campo .... (se o valor é um número inteiro, o canhão pertence ao jogador 1, caso contrário ao jog. 2).

Teremos ainda que situar algumas montanhas para complicar o problema. Colocamos uma montanha por linha aleatoriamente. O valor aleatório da coluna é calculado na linha 75. Simbolicamente, o valor 10 representa uma montanha.

A MATRIZ E A AFIXAÇÃO DO ÉCRAN

Para toda a manipulação (movimento, eliminação de um canhão, etc.) nós devemos tomar em conta dois factores: — posição na matriz e posição no écran.
Toda a modiicação na matriz deve refletir-se instantaneamente no écran.
Iremos usar uma subrotina, ou seja algumas linhas de programa que são acedidas por GOSUB e que regressam sempre ao ponto donde partiram, através da existência de um RETURN.
A e B são as coordenadas da matriz, V e H as coordenadas do écran. F$ contém o caracter que pretendemos afixar.
As linhas 1000/1005 calculam as coordenadas V, H, a partir de A e B e a linha 1010 afixa o valor de F$.
Vão-se defrontar dois jogadores. Incluímos portanto todas as fases do jogo num ciclo para alternar um com o outro. (Início do ciclo na linha 110 e final em 330).
Utilizamos o comando Input para pedir ao jogador as coordenadas do canhão que quer deslocar. O pedido é repetido nos seguintes casos:

a) as coordenadas estão fora de limite.
b) a casa não contém canhão.
c) o jogador actual é o 2 e a casa contém um valor inteiro (linha 136)
d) o jogador actual é o 1 e a casa contém um valor decimal (linha 137)

Quando tudo vai bem, pede-se ao jogador a direcção do movimento, guardando-a na variável M. Verificar se é um número entre 1 e 4 ......
Depois calculámos (143/148) as coordenadas da casa de chegada do tiro, A1 e B1, impedindo um movimento que sai fora do campo de jogo.
Último controlo: A casa de chegada está vazia? ...... se não, então o jogador cometeu um erro e paga-o caro .... perde o seu tiro e passa o comando ao adversário. (GOTO 330).
Quando o movimento é aceite, é preciso afixá-lo. Num primeiro tempo a casa de partida ficará vazia (A,B) (linha 155) e depois usamos o valor guardado em U (A, B) para determinar o valor a afixar... um 2 ou um 3 ou um 4, em inverso vídeo ou não.
Na linha 180, transferimos o valor do canhão para a casa de chegada, colocamos a casa de partida em zero e convertemos A em A1 e B em B1.

O TIRO EM 4 DIRECÇÕES

A subrotina de tiro está situada na linha 500. É um pouco complexa, mas vamos tentar compreender...
O tiro parte nas quatro direcções e num comprimento igual à potência da peça (a menos que encontre uma montanha).
Linha 500, colocamos FL=0, o que quer significar uma «bandeira» que indica se uma montanha foi ou não encontrada (FL=1). Se existe a eliminação de uma peça então é preciso afixar um ponto (ajustamos F$).
Nós trabalhamos com base em A e B, para os não perdermos vamos usar um depósito transitório em W e Z. Lancemos então um ciclo com índice C que irá de 1 até à potência de tiro do canhão ... (parte inteirta de U(A,B).
A cada tiro de canhão, juntamos I ao valor de A e o valor de L a B. As quatro combinações de valores de I e L farão explorar sistematicamente cada casa de direcção possível do tiro até à potência máxima.
A menos que o tiro saia do campo (502) ou que encontre uma montanha (503). Todas as casas encontradas sAo colocadas em zero (destruição) na linha 5050, com uma excepção ... se é um canhão do mesmo campo que é atingido (teste da linha 504).
Se o jogador 1 que atira (K = 1) o valor do canhão de partida é um número inteiro. Caso contrário é um número fraccionário. Em resumo se a subtracção CANHÃO QUE ATIRA — CANHÃO ENCONTRADO = NÚMERO INTEIRO, significa que é um amigo, e não se matam os amigos! (GOTO 506).
Se FL = 1 então encontrou uma montanha e também salta a linha de destruição.
O canhão atirou e então passa para o jogador 2 (K = 2), quando ele jogou passamos a linha 340 e o contador de tiros e incrementado. Quando os tiros atingem o valor 40, passamos a linha 2000 para calcular resultados.
O Programa merece ser melhorado, e a cada um os seus méritos!

```
   1 2 3 4 5 6 7 8 9     Jogo 1
:
1  2 3 4 2 3 4 2 3 4
:
2  . . . . . . . . M  Vert Horiz
:                     __________
3  . . . M . . . . .
:
4  . . . . . . . . M
:
5  . M . . . . . . .     Dir.=
:
6  . . . . . M . . .
:
7  . . . M . . . . .
:
8  . . M . . . . . .
:                        Tiro 1
9  2 3 4 2 3 4 2 3 4
```

```
  1 BORDER 0: PAPER 0: INK 7
  5 CLS
 10 DIM u(9,9)
 12 LET tiro=1
 15 FOR i=1 TO 9
 16 FOR z=1 TO 9
 17 LET u(i,z)=0
 18 NEXT z
 19 NEXT i
 20 PRINT AT 1,3;"1 2 3 4 5 6 7 8 9"
 25 FOR i=1 TO 9
 30 PRINT ":": PRINT i;" . . . . . . . . ."
 32 NEXT i
 50 PRINT AT 3,3;"2 3 4 2 3 4 2 3 4"
 52 PRINT AT 19,3;"2 3 4 2 3 4 2 3 4"
 60 LET i=1: LET v=2: LET h=3: LET z=4
 62 LET u(i,1)=v: LET u(i,2)=h: LET u(i,3)=z
 65 LET u(i,4)=v: LET u(i,5)=h:
```

```
 LET u(i,6)=z
  66 LET u(i,7)=v: LET u(i,8)=h:
 LET u(i,9)=z
  67 IF i=9 THEN GO TO 70
  68 LET i=9: LET v=2.1: LET h=3
.1: LET z=4.1: GO TO 62
  70 FOR a=2 TO 8
  75 LET b=INT (9*RND+1): LET u(
a,b)=10
  80 LET f$="M": GO SUB 1000
  90 NEXT a
 100 PRINT AT 5,21;"Vert Horiz"
 102 PRINT AT 6,21;"___________"
 104 PRINT AT 11,24;"Dir.="
 110 FOR k=1 TO 2
 115 PRINT AT 1,24;"Jogo ";k
 120 PRINT AT 18,24;"Tiro ";Tiro
 125 PRINT AT 21,0;"": INPUT b,a
: PRINT AT 6,21;"___________": P
RINT AT 6,23;b;AT 6,29;a
 130 IF a<1 OR a>9 OR b<1 OR b>9
 THEN GO TO 125
 135 IF u(a,b)<1 OR u(a,b)>4.1 T
HEN GO TO 125
 136 IF u(a,b)=INT (u(a,b)) AND
k=2 THEN GO TO 125
 137 IF u(a,b)<>INT (u(a,b)) AND
 k=1 THEN GO TO 125
 140 INPUT "direccao=";m
 142 IF m<1 OR m>4 THEN GO TO 14
0
 143 PRINT AT 11,30;m: LET a1=a:
 LET b1=b
 145 IF m=1 AND a>1 THEN LET a1=
a-1
 146 IF m=3 AND a<9 THEN LET a1=
a+1
 147 IF m=4 AND b>1 THEN LET b1=
b-1
 148 IF m=2 AND b<9 THEN LET b1=
b+1
 150 IF u(a1,b1)<>0 THEN GO TO 3
30
 155 LET f$=".": GO SUB 1000
 160 IF u(a,b)>2.5 THEN GO TO 16
5
 161 LET f$="2": IF k=2 THEN LET
 f$="▦"
 162 GO TO 180
 165 IF u(a,b)>3.5 THEN GO TO 17
0
 166 LET f$="3": IF k=2 THEN LET
 f$="▦"
 167 GO TO 180
 170 LET f$="4": IF k=2 THEN LET
 f$="▦"
 180 LET u(a1,b1)=u(a,b): LET u(
a,b)=0: LET a=a1: LET b=b1: GO S
UB 1000
 300 LET i=-1: LET l=0: GO SUB 5
00
 305 LET i=0: LET l=1: GO SUB 50
0
 310 LET i=1: LET l=0: GO SUB 50
0
 320 LET i=0: LET l=-1: GO SUB 5
00
 330 NEXT k
 340 LET tiro=tiro+1: IF tiro>40
 THEN GO TO 2000
 341 GO TO 110
 500 LET fl=0: LET f$=".": LET z
=a: LET w=b
 501 FOR c=1 TO INT (u(z,w))
 502 LET a=a+i: LET b=b+l: IF a<
1 OR a>9 OR b<1 OR b>9 THEN GO T
O 506
 503 IF u(a,b)=10 THEN LET fl=1
 504 IF u(a,b)-u(z,w)=INT (u(z,w
)) OR fl=1 THEN GO TO 506
 505 LET u(a,b)=0: GO SUB 1000
 506 NEXT c
 508 LET a=z: LET b=w: RETURN
1000 LET v=2*b+1
1005 LET h=2*a+1
1010 PRINT AT h,v;f$
1015 RETURN
2000 LET v=0: LET h=0: FOR i=1 T
O 9
2005 FOR z=1 TO 9
2010 IF u(i,z)<>INT (u(i,z)) THE
N LET p2=p2+INT (u(i,z)): GO TO
2020
2015 IF u(i,z)<0 AND u(i,z)<10 T
HEN LET p1=p1+u(i,z)
2020 NEXT z: NEXT i
2030 PRINT "jogador 1 ";p1
2040 PRINT "jogador 2 ";p2
2050 STOP
2100 REM fim do jogo
```

AS VARIÁVEIS DO PROGRAMA

U(9,9)	.....	MATRIZ DO CAMPO DE JOGO
F$	.....	GUARDA O CARACTER A FIXAR
TIRO	.....	CONTADOR DE TIROS
I,Z,W	.....	VARIÁVEIS CONTADORAS DOS CICLOS
A,B,	.....	COORDENADAS DA MATRIZ
V,H	.....	COORDENADAS DO ÉCRAN
Ai,Bi	.....	COORDENADAS (MATRIZ) DA CASA DE CHEGADA DE UM MOVIMENTO
M	.....	DIRECÇÃO DO MOVIMENTO
L	.....	AFECTA À ROTINA DE TIRO D COORDENADA B
K	.....	JOGO EM CURSO
FL	.....	BANDEIRA QUE ASSINALA SE FOI ENCONTRADA UMA MONTANHA

COMENTÁRIOS DO PROGRAMA:

5 LIMPEZA DO ÉCRAN
20/55 TRAÇAR O CAMPO DE JOGO
70/90 COLOCAR ALEAT. AS MONTANHAS
110 CICLO INICIAL DO JOGO
120 ÉCRAN DO TIRO CORRENTE
130 VALIDAR AS COORDENADAS
136/7 PERTENCE O CANHÃO AO JOGADOR?
142 VALIDAR A DIRECÇÃO
A1, B1
150 CASA VAZIA?
160/70 DETERMINAR O CARACTERE A ESCREVER

300/20 TIRO NAS 4 DIRECÇÕES PRECEDENTES
330 FIM DO CICLO
500 ROTINA DO TIRO
508 RECUPERAR A, B
2000 CÁLCULO DE RESULTADOS
10/19 INICIALIZAR A MATRIZ
60/68 TIPOS DE CANHÃO (MATRIZ)
100/04 INPUTS FUTUROS
115 ÉCRAN DO JOGAD. CORRENTE
125 INPUT DA COORD. VERT./HORIZ.
135 VERIFICAR SE EXISTE UM CANHÃO
140 INPUT DA DIRECÇÃO DO MOVIMENTO
143/8 CÁLCULO DE COORD. CHEGADA
155 LIMPAR A POSIÇÃO DE PARTIDA
180 REAJUSTAR OS VALORES DA MATRIZ APÓS O MOVIMENTO ÉCRAN DO MOV.
321/3 ELIMINAR OS INPUTS
340/1 AUMENTAR O NUM. DE TIROS
501/6 GESTÃO DE TIRO
1000 SUBROTINA DE COORDENADAS V. H

AFORRO

AFORRO — Calcula a valorização dos "Certificados de Aforro" emitidos pela Junta de Crédito Público a partir de 27/2/85 por um prazo de 5 anos em função da sua data de emissão.

```
   10 CLEAR
   11 INK 0
   12 PAPER 7
   13 BORDER 0
   14 PRINT
   15 PRINT
   20 PRINT "Calculo de
          VALORIZACAO DUM CERT
IFICADO DE": PRINT : PRINT "
      A F O R R O"
   22 PRINT
   23 PRINT
   24 PRINT
   25 PRINT "Certificados emitido
s depois de 27/2/1985"
   26 PRINT "Para continuar premi
r a letra C e depois ENTER"
   27 STOP
   35 CLEAR
   37 PRINT
   40 PRINT " INTRODUZIR A DATA D
E EMISSAO DO CERTIFICADO"
   41 GO TO 59
   42 INPUT "DIA ?";A
   43 INPUT "MES ?";B
   44 INPUT "ANO ? (4 Algarismos)
";C
   45 CLS : PRINT "INTRODUZIR A D
ATA PARA CALCULO DA VALORIZACAO"
   46 INPUT "DIA ?";D
   47 INPUT "MES ?";E
   48 INPUT "ANO ? (4 algarismos)
";F
   49 IF b>2 THEN GO TO 52
   50 LET g=365*c+a+31*(b-1)+INT
((c-1)/4)-INT (.75*(INT ((c-1)/1
00)+1))
   51 GO TO 53
   52 LET g=365*c+a+31*(b-1)-INT
(.4*b+2.3)+INT (c/4)-INT (.75*(I
NT (c/100)+1))
   53 IF e>2 THEN GO TO 56
   54 LET h=365*f+d+31*(e-1)+INT
((f-1)/4)-INT (.75*(INT ((f-1)/1
00)+1))
   55 GO TO 57
   56 LET h=365*f+d+31*(e-1)-INT
(.4*e+2.3)+INT (f/4)-INT (.75*(I
NT (f/100)+1))
   57 IF g<725064 OR h<725064 THE
N CLS : PRINT "ESTAO-ME A GOZAR
OU QUE ? - Para continuar premir
 a letra C e depois ENTER": STOP
 : GO TO 10
   58 GO TO 210
   59 INPUT "DIA ?";A
   60 INPUT "MES ?";B
   70 INPUT "ANO ? (4 algarismos)
";C
   80 CLS
   85 PRINT
   90 PRINT " INTRODUZIR A DATA P
ARA CALCULO DA VALORIZACAO"
  100 INPUT "DIA ?  ";D
  110 INPUT "MES ?  ";E
  120 INPUT "ANO ? (4 algarismos)
";F
  130 IF b>2 THEN GO TO 160
  140 LET g=365*c+a+31*(b-1)+INT
((c-1)/4)-INT (.75*(INT ((c-1)/1
00)+1))
  150 GO TO 170
  160 LET g=365*c+a+31*(b-1)-INT
(.4*b+2.3)+INT (c/4)-INT (.75*(I
NT (c/100)+1))
  170 IF E>2 THEN GO TO 200
  180 LET h=365*f+d+31*(e-1)+INT
((c-1)/4)-INT (.75*(INT ((f-1)/1
00)+1))
  190 GO TO 202
  200 LET h=365*f+d+31*(e-1)-INT
(.4*e+2.3)+INT (f/4)-INT (.75*(I
NT (f/100)+1))
  202 IF G>=725064 AND H>=725064
THEN GO TO 207
  204 CLS : PRINT "  ESTE PROGRAM
A SO E VALIDO PARA CERTIFICADOS
POSTERIORES A 27/2/85 - Para con
tinuar premir a letra C e depois
 ENTER"
  206 STOP : CLS : PRINT "INTRODU
ZIR UMA DATA VALIDA": GO TO 42
  207 CLS
  208 IF H>=726981 THEN PRINT "A
data para calculo nao pode ser p
osterior a 29/5/1990 - Para cont
inuar premir a letra C e depois
ENTER": STOP : GO TO 10
  210 LET i=INT ((h-g)/91.25)
  220 CLS
  225 PRINT
  230 PRINT " INTRODUZIR O VALOR
DE CUSTO DO CERTIFICADO"
  240 INPUT "VALOR DE CUSTO (Nume
ro inteiro)  ";J
  242 CLS
  244 LET J=J/70
  246 PRINT AT 4,1;"NUMERO DE UNI
DADES DE 70$00: "
  247 PRINT AT 5,14;J
  248 PRINT
  249 PRINT AT 7,1;"VALOR EM ";
D;"/";E;"/";F;":"
  250 IF I=0 THEN LET J=J*70: PRI
NT "   Escudos ";J
  251 IF I=1 THEN LET J=J*73.8: P
RINT "   Escudos ";J
  252 IF I=2 THEN LET J=J*77.8: P
RINT "   Escudos ";J
  253 IF I=3 THEN LET J=J*82: PRI
NT "   Escudos ";J
  254 IF I=4 THEN LET J=J*86.5: P
RINT "   Escudos ";J
  255 IF I=5 THEN LET J=J*91.2: P
RINT "   Escudos ";J
  256 IF I=6 THEN LET J=J*96.2: P
RINT "   Escudos ";J
  257 IF I=7 THEN LET J=J*101.5:
PRINT "   Escudos ";J
  258 IF I=8 THEN LET J=J*107: PR
INT "   Escudos ";J
  259 IF I=9 THEN LET J=J*112.9:
PRINT "   Escudos ";J
  260 IF I=10 THEN LET J=J*119.2:
 PRINT "   Escudos ";J
  261 IF I=11 THEN LET J=J*125.8:
 PRINT "   Escudos ";J
  262 IF I=12 THEN LET J=J*132.7:
 PRINT "   Escudos ";J
  263 IF I=13 THEN LET J=J*140.1:
 PRINT "   Escudos ";J
  264 IF I=14 THEN LET J=J*147.9:
 PRINT "   Escudos ";J
  265 IF I=15 THEN LET J=J*156.2:
 PRINT "   Escudos ";J
  266 IF I=16 THEN LET J=J*164.9:
 PRINT "   Escudos ";J
  267 IF I=17 THEN LET J=J*174.1:
 PRINT "   Escudos ";J
  268 IF I=18 THEN LET J=J*184: P
RINT "   Escudos ";J
  269 IF I=19 THEN LET J=J*194.3:
 PRINT "   Escudos ";J
  270 IF I=20 THEN LET J=J*205.2:
 PRINT "   Escudos ";J
  280 PRINT AT 15,5;"Outro calcul
o ? s/n"
  285 IF NOT (INKEY$="s" OR INKEY
$="n") THEN GO TO 285
  290 IF INKEY$="s" THEN GO TO 10
  300 IF INKEY$="n" THEN STOP
```

CÁLCULO DE BOBINAS DE CARGA PARA ANTENAS

Adapt. (Nuova Elettronica - 1)

```
   5 REM CALCULO DE BOBINAS DE
         CARGA PARA ANTENAS
       (NUOVA ELETTRONICA - 1)
  10 CLS : DIM A(31,15)
  20 POKE 23609,50: POKE 23658,8
  30 PRINT BRIGHT 1;"###########
#######################
                       ##    CALCU
LO DE BOBINAS DE       ##
                       ##     CARG
A PARA ANTENAS         ##
                       ############
#####################"''
 100 INPUT "% DE ENCURTAMENTO (5
 A 80)  ";L;''"% DE POS.(A/L)x10
0 (0 A 70) ";C
 120 IF L<5 OR L>80 AND C<0 OR C
>70 THEN BEEP .1,1: GO TO 100
 130 PRINT "  ENCURTAMENTO DA AN
TENA = ";L;"%"
 140 PRINT '"  POSICAO - (A/L)x1
00       = ";C;"%"
 150 LET L=(L-2.5)/2.5
 160 LET C=C/5+1
 300 FOR M=1 TO 31
 310 FOR N=1 TO 15
 320 READ A(M,N)
 330 NEXT N
 340 NEXT M
 350 PRINT
 360 BEEP .1,1: PRINT "
TABELA -   ";A(L,C);"''''''
 370 INPUT "FREQUENCIA (MHz) ";F
 375 PRINT BRIGHT 1;"  INDUTANCI
A = ";INT (A(L,C)/(2*PI*F)*100+.
05)/100;" microhenry  "''
 380 RESTORE
 390 PRINT #0; BRIGHT 1; FLASH 1
;"      MAIS  CALCULOS ?  (S/N)
  ": PAUSE 0
 400 IF INKEY$="S" THEN CLS : GO
 TO 30
 410 IF INKEY$<>"S" THEN STOP
9100 DATA 5000,5100,5400,5700,60
00,6300,6500,7000,7500,8000,8500
,9000,0000,0000,0000
9110 DATA 3900,4000,4200,4400,46
00,4800,5000,5400,5700,6200,6600
,7100,8000,0000,0000
9120 DATA 2800,2900,3000,3100,32
00,3400,3600,3800,4000,4400,4800
,5200,5600,6300,7000
9130 DATA 2450,2550,2650,2750,28
50,3025,3200,3370,3560,3900,4250
,4600,4950,5600,6250
9140 DATA 2100,2200,2300,2400,25
00,2650,2800,2950,3120,3400,3700
,4000,4300,4900,5500
9150 DATA 1750,1850,1950,2050,21
50,2275,2400,2525,2680,2900,3150
,3400,3650,4200,4750
9160 DATA 1400,1500,1600,1700,18
00,1900,2000,2100,2250,2400,2600
,2800,3000,3500,4000
9170 DATA 1290,1375,1460,1550,16
50,1740,1840,1940,2070,2220,2390
,2600,2800,3250,3700
9180 DATA 1170,1250,1320,1410,14
90,1570,1670,1770,1880,2030,2160
,2350,2600,3000,3400
9190 DATA 1060,1125,1180,1260,13
00,1400,1500,1600,1700,1850,1950
,2120,2400,2750,3100
9200 DATA 950,1000,1050,1120,118
0,1250,1340,1430,1520,1670,1730,
1900,2200,2500,2800
9210 DATA 885,930,980,1040,1100,
1160,1240,1330,1420,1550,1630,17
80,2030,2320,2600
9220 DATA 820,860,900,960,1010,1
070,1140,1210,1310,1430,1510,165
0,1850,2120,2400
9230 DATA 755,790,830,880,930,98
0,1040,1100,1200,1300,1400,1520,
1670,1900,2200
9240 DATA 690,725,760,800,850,90
0,950,1000,1100,1200,1300,1400,1
500,1750,2000
9250 DATA 650,675,710,750,780,84
0,890,940,1030,1130,1230,1330,14
40,1660,1880
9260 DATA 590,620,655,690,725,77
5,825,875,950,1050,1150,1250,137
0,1560,1750
9270 DATA 540,575,600,630,660,71
0,760,810,870,970,1050,1180,1300
,1450,1600
9280 DATA 500,525,550,575,600,65
0,700,750,800,900,1010,1130,1250
,1370,1500
9290 DATA 470,490,510,540,570,61
0,650,700,750,840,950,1050,1170,
1300,1450
9300 DATA 440,450,470,500,525,56
0,600,650,700,775,855,965,1070,1
230,1400
9310 DATA 410,420,430,460,480,51
0,550,600,650,700,750,880,980,11
60,1350
9320 DATA 380,390,400,425,450,47
5,500,550,600,650,700,800,900,11
00,1300
9330 DATA 350,360,370,390,420,44
0,470,510,550,600,650,740,840,10
00,1180
9340 DATA 320,330,340,360,380,40
0,430,460,500,550,660,680,765,90
0,1050
9350 DATA 290,300,310,330,350,37
0,390,420,450,500,550,620,690,80
0,920
9360 DATA 260,274,288,300,320,34
0,360,380,400,450,500,560,630,70
0,800
9370 DATA 235,250,260,270,290,30
5,330,355,375,430,475,530,590,65
0,765
9380 DATA 210,220,230,240,257,27
0,295,325,350,390,445,490,545,60
0,730
9390 DATA 185,190,200,210,225,23
5,260,290,325,370,410,455,500,55
0,690
9400 DATA 160,168,176,184,192,20
0,230,270,300,345,385,425,460,50
0,660
9900 STOP
9990 SAVE *"ANTENAS" LINE 1
```

«DESEJO TROCAR IDEIAS E EXPERIÊNCIAS SOBRE O PROGRAMA **MEGABASIC** E SOBRE ROTINAS PARA INTRODUÇÃO DE CÓDIGO MÁQUINA. COMPRO OU TROCO FOTOCÓPIAS DO «YOUR SPECTRUM» DE JANEIRO A MARÇO RESPEITANTES A RUBRICAS DESTE MESMO PROGRAMA **YSMEGABASIC**».

FUNÇÕES COM TRÊS RAMOS

FERNANDO MOURA
V. N. GAIA

Este programa permite desenhar gráficos de funções com três ramos.
Experimente por exemplo:

f(x)=100 × sen (x/20)
g(x)=x × x
h(x)=—x × 50

NOTA: Se o gráfico da função escrita sair do ecran, o computador desenha sobre o eixo dos xx e passa para o ramo seguinte.

```
   4 FOR x=0 TO 703: PRINT "*";:
 NEXT x
   5 PRINT AT 10,5; FLASH 1;"FUN
COES COM TRES RAMOS": PAUSE 100:
 CLS
  10 PLOT 0,70: DRAW 250,0
  20 PLOT 125,0: DRAW 0,175
  30 PLOT 80,60: DRAW 0,20
  40 PLOT 170,60: DRAW 0,20
  50 PLOT 125,175: DRAW -3,-3
  60 PLOT 125,175: DRAW 3,-3
  70 PLOT 250,70: DRAW -3,3
  80 PLOT 250,70: DRAW -3,-3
  90 PRINT AT 14,30;"x"
 100 PRINT AT 0,14;"y"
 110 INPUT "f(x)=";f$: INPUT "g(
x)=";g$: INPUT "h(x)=";h$
 140 PRINT f$: PRINT g$: PRINT h
$
 200 FOR x=-125 TO -45: LET f=(V
AL f$/10)+70
 220 IF f>175 OR f<0 THEN LET f=
70
 230 PLOT x+125,f: NEXT x
 300 FOR x=-45 TO 45: LET g=(VAL
g$/10)+70
 320 IF g>175 OR g<0 THEN LET g=
70
 330 PLOT x+125,g: NEXT x
 400 FOR x=45 TO 120: LET h=(VAL
h$/10)+70
 420 IF h>175 OR h<0 THEN LET h=
70
 430 PLOT x+125,h: NEXT x
 450 PRINT #0;"tecle para nova f
uncao"
 500 PAUSE 0: CLS : GO TO 10
 600 SAVE "ftr" LINE 0
```


DECOMPOSIÇÃO DE PALAVRAS EM SILABAS

J. M. 85

```
   5 REM DECOMPOS. DE PALAVRAS
          EM SILABAS - J.M.85

  10 CLS : DIM w$(60)
  20 DEF FN c(x$)=x$>="b" AND x$
<="z" AND x$<>"e" AND x$<>"i" AN
D x$<>"o" AND x$<>"u"
  30 DEF FN v(x$)=x$="a" OR x$="
e" OR x$="i" OR x$="o" OR x$="u"
  35 DEF FN e(x$)=x$="lh" OR x$=
"nh" OR x$="br" OR x$="cr" OR x$
="dr" OR x$="fr" OR x$="gr" OR x
$="pr" OR x$="tr" OR x$="vr"
  40 DEF FN d(x$)=x$="ae" OR x$=
"ai" OR x$="ao" OR x$="au" OR x$
="ei" OR x$="eu" OR x$="iu" OR x
$="oe" OR x$="oi" OR x$="ou" OR
x$="ua" OR x$="ue" OR x$="ui" OR
 x$="uo"
  50 INPUT "Palavra? ";a$: LET l
=LEN a$: IF l=0 THEN GO TO 50
  60 FOR i=1 TO l: LET x$=a$(i T
O i): IF CODE x$>=65 AND CODE x$
<=90 THEN LET x$=CHR$ (CODE x$+3
2)
  70 LET w$(i)=x$: NEXT i: LET i
=0
  80 LET i=i+1: IF i>=l THEN GO
TO 1000
  90 IF NOT FN v(w$(i)) THEN GO
TO 80: REM [illegible]
 100 IF NOT FN c(w$(i+1)) THEN G
O TO 180: REM [illegible]
 110 REM *** VOGAL-CONSOANTE ***
 120 IF i+2>l THEN GO TO 1000
 130 IF NOT FN c(w$(i+2)) THEN G
O TO 2000: REM [illegible]
 140 LET x$=w$(i+1)+w$(i+2): IF
FN e(x$) THEN GO TO 2000: REM [illegible]
 150 IF i+3>l THEN GO TO 1000
 160 IF w$(i+2)="s" AND FN c(w$(
i+3)) THEN LET i=i+2: GO TO 2000
: REM [illegible]
 170 LET i=i+1: GO TO 2000
 180 REM *** VOGAL-VOGAL ***
```

```
 190 IF i+2>l THEN GO TO 1000
 200 IF w$(i)=w$(i+1) THEN GO TO
 300: REM [illegible]
 210 LET x$=w$(i)+w$(i+1): IF NO
T FN d(x$) THEN GO TO 300: REM [illegible]
 220 IF FN v(w$(i+2)) THEN LET i
=i+1: GO TO 300: REM [illegible]
 225 IF i+3<l THEN LET x$=w$(i+2
)+w$(i+3): IF x$="lh" OR x$="nh"
 THEN LET i=i+1: GO TO 2000
 230 IF i+3>l THEN GO TO 1000: R
EM [illegible]
 240 IF FN v(w$(i+3)) THEN LET i
=i+1: GO TO 2000: REM [illegible]
 250 LET i=i+2: GO TO 2000: REM
[illegible]
 300 LET l=l+1: FOR k=l TO i+2 S
TEP -1: LET w$(k)=w$(k-1): NEXT
k: LET w$(i+1)="-": LET i=i+1: G
O TO 80
1000 REM [illegible]
1010 FOR i=1 TO l: PRINT w$(i);:
 NEXT i: PRINT : GO TO 50
2000 LET l=l+1: FOR k=l TO i+2 S
TEP -1: LET w$(k)=w$(k-1): NEXT
k: LET w$(i+1)="-": LET i=i+2: G
O TO 80
9990 SAVE *"SILABAS" LINE 1
```

CALENDÁRIO

Adapt. da DISK — DEMO TIMEX

MES: MAI ANO: 1985

SE	TE	QU	QU	SE	SA	DO
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

```
  5 REM         [illegible]
          ADAP.DA DISK-DEMO TIMEX
 10 BEEP .2,5
 20 CLS : PRINT BRIGHT 1;"
                              * *
 C A L E N D A R I O  * *
                              "'''
 30 PRINT TAB 3;"Este calendari
o so e valido a partir de 1582
."''
 40 PRINT TAB 3;"Isto porque
Julius Caesar sobrestimou a du
racao do ano em 11 minutos."'
'
 50 PRINT TAB 3;"A correccao fo
i feita , pelo Papa Gregorio XI
II,no referido ano."''
 60 PRINT TAB 3;"Por isso e des
ignado por ca- lendario gregori
ano."
 70 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175:
 PLOT 0,151: DRAW 255,0
 80 PRINT AT 20,7; BRIGHT 1;" P
RIMA UMA TECLA "
 90 IF INKEY$<>"" THEN GO TO 90
 100 IF INKEY$="" THEN GO TO 100
 110 BEEP .2,5: CLS
 120 RESTORE
 130 LET b$="JANFEVMARABRMAIJUNJ
ULAGOSETOUTNOVDEZ"
 140 PRINT ; BRIGHT 1;"
                          * * *
 O P C O E S  * * *
                          "'''
 150 PRINT TAB 2; BRIGHT 1;"1";
BRIGHT 0;" - CALENDARIO DE UM
 MES"''
 160 PRINT TAB 2; BRIGHT 1;"2";
BRIGHT 0;" - DIFERENCA.EM DIAS,E
NTRE "'';TAB 6;"DUAS DATAS"''
 170 PRINT TAB 2; BRIGHT 1;"3";
BRIGHT 0;" - CALCULO DO DIA DA S
EMANA"'';TAB 6;"DE UMA DADA DATA
"''
 175 PRINT TAB 2; BRIGHT 1;"4";
BRIGHT 0;" - TERMINAR"
 180 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175:
 PLOT 0,151: DRAW 255,0
 190 PAUSE 10
 200 LET a$=INKEY$: IF a$="" OR
a$>CHR$ (52) OR a$<CHR$ (49) THE
N GO TO 200
 210 IF a$="2" THEN GO TO 530
 220 IF a$="3" THEN GO TO 585
 225 IF a$="4" THEN STOP
 230 CLS : BEEP .2,5
 240 PRINT AT 1,4; BRIGHT 1;" ME
S:"; BRIGHT 0;AT 1,16; BRIGHT 1;
" ANO:"; BRIGHT 0
 250 FOR d=6 TO 24 STEP 3
 260 READ a$: PRINT AT 4,d-1; BR
IGHT 1;a$
 270 NEXT d
 280 FOR h=0 TO 112 STEP 16: PLO
T 35,147-h: DRAW 168,0: NEXT h
 290 FOR v=0 TO 168 STEP 24: PLO
T 35+v,147: DRAW 0,-112: NEXT v
 300 LET d=1: GO SUB 660
 310 PRINT AT 1,10;b$(m1 TO m1+2
)
 320 PRINT AT 1,22;q
 330 LET dw=f-7*INT (f/7)
 340 LET d=1: GO SUB 700: LET f1
=f
 350 LET m=m+1: IF m>12 THEN LET
 q=q+1: LET m=1
 360 GO SUB 700: LET e=f-f1
 370 LET i=0: LET n=0
 380 LET p=21+dw*3
 390 IF p>24 THEN LET p=p-21
 400 FOR s=1 TO e
 410 LET n=p+n
 420 LET p=3
 430 PRINT AT 6+i,n;s
 440 IF s=9 THEN LET n=n-1
 450 IF n>22 THEN LET i=i+2: LET
 n=n-21
 460 NEXT s
 470 PAUSE 50
 480 PRINT #0; BRIGHT 1;"  PRIMA
 UMA TECLA PARA OPCOES  "
 490 IF INKEY$<>"" THEN GO TO 11
0
 500 GO TO 490
 510 PAUSE 100
```

```
 520 STOP
 530 CLS : BEEP .2,5: PRINT TAB
4;"DIFERENCA,EM DIAS,ENTRE"'';TA
B 10;"DUAS DATAS"
 532 PRINT AT 6,2; BRIGHT 1;" IN
TRODUZA A PRIMEIRA DATA "
 535 GO SUB 650: LET f1=f
 540 LET dw=f-(INT (f/7))*7
 550 PRINT AT 6,2; BRIGHT 1;" IN
TRODUZA A SEGUNDA DATA ": GO SUB
 650: LET f2=f: CLS
 560 LET dw=f-7*INT (f/7)
 570 LET dif=f2-f1: BEEP .2,5: P
RINT ': PRINT "A DIFERENCA ENTRE
 AS DUAS DATAS"''"          E DE ";
dif;" DIAS"
 580 GO TO 480
 585 CLS : BEEP .2,5: PRINT TAB
4;"CALCULO DO DIA DA SEMANA "'';
TAB 10;"DE UMA DATA"''''
 590 GO SUB 650: LET dw=f-7*INT
(f/7)
 600 CLS : BEEP .2,5: PRINT AT 6
,6;"O DIA DA SEMANA DE";AT 8,6;d
;" DE ";b$(m1 TO m1+2);" DE ";q;
" E"
 610 PRINT AT 10,12; BRIGHT 1;"S
ABADO DOMINGOSEGUNDA TERCA QUART
A QUINTA  SEXTA "((dw*7)+1 TO (d
w*7)+7)
 620 GO TO 480
 630 STOP
 640 DATA "SE","TE","QU","QU","S
E","SA","DO"
 650 INPUT "INTRODUZA O DIA DO M
ES :"'';d
 660 INPUT "INTRODUZA O MES :
"'';m
 670 INPUT "INTRODUZA O ANO NA F
ORMA (YYYY):       "'';q
 680 IF m>12 OR m<1 OR d<1 OR d>
31 THEN PRINT AT 19,1;"NAO BRINQ
UE !": PAUSE 60: GO TO 110
 690 LET m1=((m-1)*3)+1
 700 IF m<3 THEN LET f=365*q+d+3
1*(m-1)+INT ((q-1)/4)-INT (.75*(
INT ((q-1)/100))+1): RETURN
 710 LET f=365*q+d+31*(m-1)-INT
(.4*m+2.3)+INT (q/4)-INT (.75*IN
T (q/100)+1)
 720 RETURN
9990 SAVE "CALENDARIO" LINE 1
```

TOTOLOTO

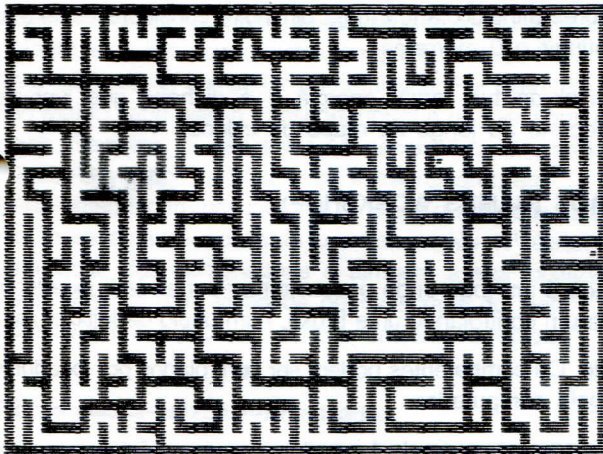
FERNANDO MACHADO
Oliveira de Azeméis

Este programa simula uma grelha do 'TOTOLOTO' e apresenta palpites para apostas simples ou multiplas, usando numeros aleatorios.

```
   0>REM      ▒▒▒ TOTOLOTO ▒▒▒

       ©1985 Fernando Machado

 Programa feito em 29/04/1985
   5 CLS : INK 0: LET f$=" Simp
les ou Multiplas?    ": PRINT AT
 0,0;
  10 RANDOMIZE : PRINT PAPER 2;"

"
  20 PRINT PAPER 2;"         "; BRI
GHT 1; PAPER 7;"  T O T O L O T
O  "; BRIGHT 0; PAPER 2;"
"
  30 PRINT PAPER 2;"
                    "
  40 DIM n(12)
  50 LET n=0: LET d$=""
  90 INK 2: PLOT 0,151: DRAW 0,-
151: DRAW 255,0: DRAW 0,151
  99 PLOT 20,28: DRAW 0,110: DRA
W 216,0: DRAW 0,-110: DRAW -216,
0
 100 INK 2: PAPER 7: PRINT AT 5,
7;" 1   2   3   4   5   6"
 110 PRINT AT 7,3;" 7   8   9   1
0  11  12  13"
 120 PRINT AT 9,3;"14  15  16  1
7  18  19  20"
 130 PRINT AT 11,3;"21  22  23
24  25  26  27"
 140 PRINT AT 13,3;"28  29  30
31  32  33  34"
 150 PRINT AT 15,3;"35  36  37
38  39  40  41"
 160 PRINT AT 17,3;"42  43  44
45"
 162 IF NOT n THEN GO SUB 2000
 163 IF d$="m" THEN LET n=0
 164 LET n=n+1: IF n>10 THEN LET
 n=1
 165 IF NOT o THEN LET d$=""
 167 PRINT AT 5,3;"  ": PRINT AT
5,3; PAPER 2; INK 7;n
 200 FOR f=1 TO num
 210 LET n(f)=1+INT (RND*45)
 220 GO SUB 500
 225 GO SUB 1000
 240 NEXT f
 250 GO SUB 2000: GO TO 100
 500 FOR g=1 TO f-1
 510 IF n(f)=n(g) THEN GO TO 210
 520 NEXT g
 530 RETURN
1000 LET x=5: IF n(f)<7 THEN GO
TO 1100
1020 LET x=7: IF n(f)<14 THEN GO
 TO 1110
1030 LET x=9: IF n(f)<21 THEN GO
 TO 1130
1040 LET x=11: IF n(f)<28 THEN G
O TO 1140
1050 LET x=13: IF n(f)<35 THEN G
O TO 1150
1060 LET x=15: IF n(f)<42 THEN G
O TO 1160
1070 LET x=17: GO TO 1170
1100 LET y=4*n(f)+3: GO TO 1200
1120 LET y=4*n(f)-25: GO TO 1200
1130 LET y=4*n(f)-53: GO TO 1200
1140 LET y=4*n(f)-81: GO TO 1200
1150 LET y=4*n(f)-109: GO TO 120
0
1160 LET y=4*n(f)-137: GO TO 120
0
1170 LET y=4*n(f)-165
1200 LET a$=STR$ n(f)
1210 IF LEN a$=1 THEN LET a$=" "
+a$
1220 PRINT AT x,y; PAPER 0; INK
6;a$
1230 RETURN
2000 INK 0: PRINT AT 20,3;f$
2002 LET g$=INKEY$
2005 IF g$="s" THEN LET num=6: L
ET o=0: RETURN
2010 IF g$="m" THEN LET d$="m":
```

```
LET o=1: GO TO 2030
2020 GO TO 2000
2030 PRINT AT 20,3;"7(Q)8(W)9(E)
10(R)11(T)12(Y)"
2040 IF INKEY$="q" THEN LET num=
7: RETURN
2045 IF INKEY$="w" THEN LET num=
8: RETURN
2050 IF INKEY$="e" THEN LET num=
9: RETURN
2055 IF INKEY$="r" THEN LET num=
10: RETURN
2060 IF INKEY$="t" THEN LET num=
11: RETURN
2065 IF INKEY$="y" THEN LET num=
12: RETURN
2070 GO TO 2030
3000>SAVE "TOTOLOTO" LINE 0
```

LABIRINTOS

Adapt. "ZX COMPUTING" DEC/JAN. 85

```
  5 REM    MAZES - ZX COMPUTING
                  DEC/JAN85
 20 CLS
 30 PRINT INVERSE 1; BRIGHT 1;"
################################
#                              #
#    L A B I R I N T O S       #
#                              #
################################
"
 40 PRINT #0; BRIGHT 1;"  * E
M   C A L C U L O   *  "
 50 LET w=31: LET h=19: GO TO 9
0
 60 LET px=xx*4-4: LET py=yy*4+
12
 70 OVER 1: FOR z=0 TO 3: PLOT
px+z,py: DRAW 0,3: NEXT z: OVER
0
 80 RETURN
 90 LET r=2*w+1: LET s=2*h+1: I
F r>63 OR s>39 OR r<10 OR s<10 T
HEN PRINT AT 0,0;"      COMPRIMENT
O OU LARGURA              out of
 range": GO TO 500
100 DIM a(r,s)
110 DIM d(4,24): RESTORE 440: F
OR i=1 TO 24: FOR j=1 TO 4: READ
 d(j,i): NEXT j: NEXT i
120 DIM i(4): DIM j(4): DIM k(4
): DIM l(4): FOR i=1 TO 4: READ
i(i): READ j(i): READ k(i): READ
 l(i): NEXT i
130 FOR i=1 TO r: FOR j=1 TO s:
 LET a(i,j)=-1: NEXT j: NEXT i
140 FOR i=2 TO r-1 STEP 2: FOR
j=2 TO s-1 STEP 2: LET a(i,j)=0:
 NEXT j: NEXT i
150 CLS
160 FOR x=1 TO r STEP 2: FOR z=
0 TO 3: PLOT (x-1)*4+z,16: DRAW
0,4*s-1: NEXT z: NEXT x
170 FOR y=1 TO s STEP 2: FOR z=
0 TO 3: PLOT 0,4*y+12+z: DRAW 4*
r-1,0: NEXT z: NEXT y
180 RANDOMIZE : LET x=2*INT (RN
D*(r-1)/2)+2: LET y=2*INT (RND*(
s-1)/2)+2: LET a(x,y)=-9: LET x1
=x: LET y1=y: LET v=1: LET max=2
0: LET done=1
190 PLOT x1*4-3,y1*4+13: DRAW 1
,0: DRAW 0,1: DRAW -1,0
200 LET c=1
210 LET d=INT (RND*23)+1
220 LET vx=x+k(d(c,d)): LET vy=
y+l(d(c,d))
230 IF vx=0 OR vx=r+1 OR vy=0 O
R vy=s+1 THEN GO TO 310
240 IF a(vx,vy)<>0 THEN GO TO 3
10
250 LET v=v+1: LET a(vx,vy)=v:
LET done=done+1: LET a(x+i(d(c,d
)),y+j(d(c,d)))=0
260 LET xx=x+i(d(c,d)): LET yy=
y+j(d(c,d)): GO SUB 60
270 IF done=h*w THEN GO TO 380
280 LET x=vx: LET y=vy: LET c=1
290 IF v>max THEN LET max=v: LE
T x9=x: LET y9=y
300 GO TO 210
310 IF c<4 THEN LET c=c+1: GO T
O 220
320 LET v=v-1
330 FOR i=1 TO 4: LET vx=x+k(d(
i,d)): LET vy=y+l(d(i,d)): IF vx
=0 OR vy=0 OR vx=r+1 OR vy=s+1 T
HEN GO TO 350
340 IF a(vx,vy)=v THEN GO TO 37
0
350 NEXT i
360 GO TO 380
370 LET x=vx: LET y=vy: GO TO 2
00
380 PLOT x9*4-3,y9*4+13: DRAW 1
,0: DRAW 0,1: DRAW -1,0
390 PRINT #0; BRIGHT 1;"   C
OPIA IMPRESSA (S/N)        "
400 PAUSE 0
410 IF INKEY$="s" OR INKEY$="S"
THEN COPY
420 IF INKEY$="n" OR INKEY$="N"
THEN RUN
430 GO TO 50
440 DATA 1,2,3,4,1,2,4,3,1,3,2,
4,1,3,4,2,1,4,2,3,1,4,3,2
450 DATA 2,1,3,4,2,1,4,3,2,3,1,
4,2,3,4,1,2,4,1,3,2,4,3,1
460 DATA 3,1,2,4,3,1,4,2,3,2,1,
4,3,2,4,1,3,4,1,2,3,4,2,1
470 DATA 4,1,2,3,4,1,3,2,4,2,1,
3,4,2,3,1,4,3,1,2,4,3,2,1
480 DATA 0,1,0,2,1,0,2,0,0,-1,0
,-2,-1,0,-2,0
9990 SAVE *"MAZES" LINE 1
```

NOVOS PROGRAMAS

GIFT FROM THE GODS (Ocean)

Até os deuses mitológicos são motivo de inspiração para os programadores.
Os deuses Apollo e Zeus deram-te Orestes, uma espada mágica para te ajudarem na tua missão através do labirinto, por baixo do palácio de Mycernea. Se fores bem sucedido ocuparás o trono, que por direito é teu.

Para sobreviveres e completares a tua missão, tens que juntar as seis figuras Euclidianas e colocá-las correctamente no quarto do guardião. A tua irmã, Electra, sabe onde se encontram as figuras mas, infelizmente, encontra-se perdida no labirinto.

A tua missão será complicada por semi-deuses que habitam o labirinto e projectam imagens de si para te confundirem. Também projectam imagens de monstros, que te tiram a força, mas que podem ser mortos com a espada que os deuses te deram.

O teu maior inimigo é a tua mãe, que matou o teu padrasto, o Rei e que deseja matar-te a ti e à tua irmã, pois o seu principal objectivo é clamar o trono para si própria.

Este jogo é representado por personagens animadas, podendo Orestes ser controlado com Joystick ou através do teclado.

O jogo utiliza um «Joystick inteligente» — a utilização deste pode ser interpretada de diversas maneiras dependendo daquilo que o programa pensar que tu pretendes fazer.

Orestes é um ser humano; por isso, as suas acções enfraquecem-no. Sempre que caíres ficarás fraco; se tiveres que voar sobre os buracos que aparecem no labirinto isso também te tirará as forças. A maneira de ficares novamente em forma é encontrares rapidamente o quarto do guardião que possui poderes que te farão recuperar imediatamente as energias perdidas, para que possas concretizar o teu objectivo com sucesso.
Boa sorte.

CITY OF DEATH (Red Shift)

O nome de «City of Death» (Cidade da Morte) é realmente apropriado. Tu és um estranho na cidade. A cidade tem uma polícia que se ocupa dos estranhos — caça-os, mata-os ou prende-os.

Embora sejas conhecido como um grande esgrimista e a tua missão será encontrares o mágico Bellatrix, terás que usar a tua espada para o localizares. Para sobreviveres na cidade, terás que assaltar casas para obteres comida e dinheiro, com o qual poderá comprar feitiços. Não sejas muito duro com os habitantes da cidade, pois eles sabem onde habita o feiticeiro.

O écran dá-te uma visão de diversas coisas:
— Um mapa da cidade com instruções ao lado deste.
— Por baixo do mapa tens uma figura contigo e com o adversário com quem estás a lutar na altura. A alternativa à luta é renderes-te e ficares preso durante 30 dias. Se lutares, tens que escolher onde queres atingir o teu adversário. A cabeça é mais difícil de atingir que o corpo. No entanto, se deres uma boa pancada na cabeça do teu adversário provocarás mais danos.

Se fores atingido, a tua força diminuirá consideravelmente e se atingir o 0 morrerás.

O nível da tua força, energia e o dinheiro que juntaste aparece-te ao longo do mapa da cidade.

CHAOS (GAMES WORKSHOP)

Chaos é um jogo onde podem entrar cerca de 8 jogadores, cada um controlando uma bruxa, num duelo de feitiços até à morte. Os combatentes podem ser controlados por jogadores ou pelo computador.

Num jogo com mais de 1 jogador, os outros não devem olhar para o écran sempre que não for a vez deles de jogar. Um jogador pode escolher um dos seus feitiços depois da selecção inicial de 12 feitiços para cada jogador.

Esta selecção é feita entre cerca de 50 feitiços, o que garante um jogo diferente sempre que o jogares. Quando cada bruxa escolher o seu feitiço, o jogo prossegue até ao écran de combate. Aqui os resultados dos feitiços são determinados depois do movimento das feiticeiras e das suas criações à volta da arena.

Depois das feiticeiras se moverem e depois de alguns inimigos terem sido abatidos, elas escolhem os feitiços para o próximo combate. O jogo continuará até que apenas uma bruxa permaneça na arena.

Os feitiços são bastante diversos: leões, crocodilos, gigantes, dragões, ursos, etc.

No entanto, existem feitiços mais poderosos como: espectros, vampiros, fantasmas . . .

Obviamente que, com todos estes perigos, as bruxas tentarão proteger-se, utilizando feitiços como: espadas mágicas, asas, armaduras, escudos, etc. Também criarão castelos, bosques para se esconderem.

Apesar de toda esta diversidade de feitiços, existe um que os supera a todos: a justiça que, com um simples sopro, tudo derrota.

Também existem feitiços ilusórios que aparecem na arena como sendo reais e que têm imenso sucesso. Estes só desaparecem depois que alguma bruxa os descubra.
Portanto, as feiticeiras podem usar ilusões para confundirem os outros jogadores.

SHADOWFIRE (BEYOND)

É uma aventura, mas com a particularidade de não ter texto. O habitual texto, muito confuso por vezes, é substituído por imagens. Por exemplo, em vez de premires «USE THE . . .», simplesmente escolhes a imagem que mostra um dedo premindo um botão.
Estas personagens possuem características criminosas. Desde ladrões até assassinos, elas constituem o Grupo Enigma.
O objectivo do jogo é, com este grupo, invadires a nave do diabólico general Zoff e salvares um embaixador que foi raptado.
Cada personagem deste jogo tem as suas próprias ideias e objectivos, portanto, deves manter-te atento a todas elas. No entanto, elas são especialistas em qualquer coisa importante para que possas completar a tua missão. Por exemplo, apenas Manto sabe usar o transportador que os transporta até à nave; Severina, a assassina, é especialista em arrombar portas; Syylk possui uma força extraordinária, etc.
Cada personagem possui força, energia e agilidade, que estão representadas no écran por barras. No 1.º écran tens uma visão das personagens e também a sua localização. Tens também uma visão de outros écrans do jogo.
Esta aventura é fácil de jogar e bastante divertida.

Filtros passa-baixos normalizados **720$**

Filtros BUTTERWORTH, de cinco elementos, utilizáveis em Baixa-Frequência e Alta-Frequência, de baixa e alta impedância. Pode funcionar com outro número de elementos e valores normalizados.

Linhas de transmissão, coaxiais e paralelas **960$**

Calcula a impedância de linhas de transmissão, coaxiais e paralelas, a partir de espaços, diâmetros dos conductores internos e externos, de centro a centro. Para comprimentos eléctricos (angulares) entrar com o coeficiente dialéctrico, e frequência.

Conversão de unidades eléctricas **720$**

Converte diversas unidades eléctricas, em corrente alterna: VOLTS/DBM, WATTS/VOLTS, WATTS/DBM, DBM/VOLTS e DBM/WATTS.

Relação de ondas estacionárias **720$**

A partir de uma dada potência aplicada, determina, por saltos a estipular, num máximo SWR admitido, a potência transmitida e a potência reflectida.

Antenas — ângulo de fogo **1080$**

Considera um dipolo de l/2=meia onda, e a frequência de ressonância. Determina o ângulo de elevação, relativo a uma perpendicular à tangente sobre o horizonte, assim como a respectiva altura ao solo.

Antenas Parabólicas **1080$**

Determina o ganho de potência, em DB., largura do feixe a meia potência (3 DB.), dos refletores parabólicos, em função do diâmetro.
Calcula também a distância focal. Considera o factor de iluminação =.5, que pode ser modificado em função das características do reflector em análise.

Transformadores **720$**

Alimentação, baixa e alta-tensão.

Circuítos sintónicos **960$**

De audio-frequência a rádio-frequência, na associação de L e C, determina a frequência de ressonância, onde c = l.

Filtros activos passa-banda **720$**

Utilizável onde se pretenda uma reduzida banda passante, cuja frequência central será condicionada pelo o I.C. utilizado. Em frequência de audio podem ser utilizados I.C. 741, LM 301 ou 747. Também podem ser associadas duas, ou mais, células sob risco de realimentação.

Ponto de admitâncias **960$**

Calcula o valor das admitâncias entre xx e 50 mHz para os valores de C e L e as diferenças de potencial aos terminais.

Atenuadores T e H **720$**

Calcula quadrípolos, simétricos e assimétricos, como malhas de atenuação resistiva, para baixa e alta frequência, considerando que R = R jx.

Rede de adaptação de impedâncias **720$**

Determina o valor dos componentes resistivos de malhas de adaptação de impedâncias, com a menor atenuação.

Antenas-Propagação-Dipolos-Proj. Polar e Cartesiana **2400$**

Calcula antenas dipolo, horizonte rádio, potência de recepção, atenuação no espaço livre, transformadores de impedância em antenas dipolo, intensidade de campo e projecção polar e cartesiana.

CLUBE Z80

INSCRIÇÃO COMO ASSOCIADO

O **CLUBE Z80** está aberto a todos os utilizadores de microcomputadores.

A intenção de associar os entusiastas das micro-máquinas, é exclusivamente a de permitir:

1 — PUBLICAÇÃO DE UM JORNAL MENSAL, onde sejam publicados programas de uso geral ou específico como no caso da educação.

2 — PROMOVER TROCAS DE PROGRAMAS, e trocas de experiências; tanto no caso do Software (programação), como no caso do Hardware (electrónica).

3 — PROMOVER DESCONTOS NA AQUISIÇÃO DE PROGRAMAS.

4 — LANÇAR CURSOS DE PROGRAMAÇÃO EM BASIC — PASCAL OU OUTRAS LINGUAGENS E DIVULGAR O USO DE LINGUAGEM MÁQUINA.

NOME ....

IDADE .... COMPUTADOR TIPO ....

PROFISSÃO ....

ENDEREÇO ....

.... TELEF. ....

ASSINATURA ANUAL — Esc. 1 500$00 ☐

ASSINATURA SEMESTRAL — Esc. 750$00 ☐

CHEQUE OU VALE DO CORREIO

N.º ....

BANCO ....

DATA ..../......./.......

JÁ SÓCIO ☐

NOVO SÓCIO ☐ → A partir do mês de .... (inclusive)